METHOD AND APPARATUS FOR VIEWING AND INTERACTING WITH A SPREADSHEET FROM WITHIN A WEB BROWSER

BACKGROUND OF THE INVENTION

With the advent and explosion of the Internet, computer users have grown accustomed to conveniently accessing virtually any kind of electronic document from virtually any location. In particular, the proliferation of the World Wide Web (the "Web") and Web browser application programs has made accessing many kinds of documents, such as text and graphics documents, very convenient. Through a Web browser application program, a user can access and view many types of electronic documents without the need for any additional software.

For some types of electronic documents, however, Web browsers are incapable of rendering the documents without the use of additional software. For example, for electronic documents that are not natively renderable by a Web browser application program, a helper or “plug-in” program must be utilized in conjunction with the Web browser application to enable viewing of the documents. One such type of document that requires the use of a plug-in to be viewable within the context of a Web browser application is the electronic spreadsheet document.

In the past, electronic spreadsheet documents (referred to herein as “spreadsheets” or “workbooks”), have not been viewable within the context of a Web browser application program without the use of a plug-in. However, the use of a plug-in can be frustrating for a computer user for a number of reasons. First, if the user is connected to the Internet via a slow connection, it may take a considerable amount of time to download and install the plug-in. Second, because plug-ins are typically written for use with a particular brand of Web browser, a plug-in may be unavailable or incompatible with the Web browser application utilized by the user. Additionally, the user may also be disappointed that in many cases, the plug-in provides only the ability to view the spreadsheet, and even when it does allow the user the ability to interact with and edit the spreadsheet, the interaction and editing environment do not match that of

the fully functioned spreadsheet application program in which the spreadsheet was authored. As an alternative to utilizing a Web browser plug-in, a user could utilize a spreadsheet application program to save a version of the spreadsheet in the hypertext markup language ("HTML") format. The user could then access the HTML version of the spreadsheet with a Web browser application. However, this method requires a user to manually save the spreadsheet file as an HTML file. Moreover, viewing the HTML version of a spreadsheet in a Web browser also does not provide the ability to interact with or edit the spreadsheet. Furthermore, viewing an HTML version of the spreadsheet using previous solutions does not provide a visual interface substantially similar to the spreadsheet application program with which the user may be familiar.

It is with respect to these considerations and others that the various embodiments of the present invention have been made.

BRIEF SUMMARY OF THE INVENTION

In accordance with the present invention, the above and other problems are solved by a method, system, and apparatus for enabling a spreadsheet, or a workbook that includes one or more spreadsheets, to be viewed and modified within the context of a Web browser application program. According to aspects of the invention, no additional software, such as a helper application or "plug-in," is required to view and interact with the spreadsheet or workbook from within the Web browser application. Moreover, when rendered in the Web browser, the spreadsheet will be displayed in a manner that is substantially similar to the spreadsheet when displayed by a spreadsheet application program.

According to one aspect of the invention, a method is provided for viewing and interacting with a spreadsheet from within the context of a Web browser application program. According to the method, a request may be received to open a spreadsheet document in the Web browser application program. In response to the request, a representation of the spreadsheet document capable of being rendered by the Web browser application program is generated. The representation of the spreadsheet document is then transmitted to the Web browser application program. The Web

browser application program then provides a viewable version of the spreadsheet document by rendering the representation of the spreadsheet document.

According to other aspects of the invention, the viewable version of the spreadsheet is rendered so that it appears substantially similar to a visual display provided by a spreadsheet application program when utilized to view the spreadsheet document. Moreover, the representation of the spreadsheet document includes only markup language tags for use in rendering the representation of the spreadsheet document along with scripts executable within the context of the Web browser for providing interactivity with the viewable version of the spreadsheet. No additional plug-ins or helper applications are required to provide interactivity with the viewable version of the spreadsheet within the Web browser.

According to other aspects of the method, a user request may be received to modify an aspect of the spreadsheet from within the Web browser. In response to the request, a script may be executed in the context of the Web browser to cause the requested modification to be performed. In particular, executing the script may include transmitting data identifying the requested modification to a server computer. The server computer may receive the data and, in response thereto, may generate a recalculated spreadsheet document that reflects the requested modification. The server computer may then generate a representation of the recalculated spreadsheet document capable of being rendered by the Web browser application program and transmit the representation of the recalculated spreadsheet document to the Web browser application program, where it is rendered.

According to other aspects of the invention, another method for viewing and interacting with a spreadsheet from within a Web browser application program is provided. According to this method, a request generated from within a Web browser application program is received for a workbook that includes one or more spreadsheets. In response to the request, the requested workbook is retrieved and calculated. The calculated workbook is then converted to the extensible markup language spreadsheet format ("XMLSS"). The XMLSS version of the workbook is then converted from the XMLSS format into a renderable document. According to various aspects of the

invention, the renderable document includes HTML and scripts capable of being rendered by the Web browser application. The scripts provide functionality for editing and interacting with the spreadsheet from within the context of the Web browser. Once the renderable document has been generated, a response is provided to the request for a workbook by providing the renderable document.

According to other aspects of the method, the renderable document may include only markup language tags for use in rendering the representation of the workbook and scripts executable within the context of the Web browser for providing interactivity with the viewable version of the workbook. Moreover, according to other aspects of the invention, the renderable document is rendered within the Web browser so that it appears substantially similar to a visual display provided by a spreadsheet application program when it is utilized to view the requested workbook.

According to yet other aspects of the invention, a system is provided for allowing a spreadsheet to be viewed and modified from within the context of a Web browser application program. The system includes a first server computer operative to execute program code for receiving a request for a renderable version of a spreadsheet from the Web browser application program. The first server computer is also operative to request from a second server computer a version of the spreadsheet expressed using the XMLSS format. When the first server computer receives the XMLSS version of the spreadsheet, it is also operative to convert the XMLSS version of the spreadsheet to a renderable version of the spreadsheet, and to respond to the request from the Web browser with the renderable version of the spreadsheet. The renderable version of the spreadsheet document includes only markup language tags for use in rendering the spreadsheet document and scripts executable within the Web browser for providing interactivity with the viewable version of the spreadsheet. Moreover, the renderable version of the spreadsheet appears substantially similar to a visual display provided by a spreadsheet application program when utilized to view the native version of the spreadsheet.

The system also includes a second server computer operative to receive requests from the first server computer for the XMLSS version of the spreadsheet. In

response to such a request, the second server computer is further operative to retrieve a native version of the spreadsheet from a database, to convert the native version of the spreadsheet to an XMLSS version of the spreadsheet, and to respond to the request from the first server computer with the XMLSS version of the spreadsheet.

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 SEVERAL VIEWS 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;

FIGURES 3 and 6 are process diagrams showing aspects of an illustrative process for enabling the viewing and editing of a spreadsheet document from within the context of a Web browser; and

FIGURES 4-5 and 7-9 are screen diagrams illustrating display screens provided by the various aspects of the invention.

DETAILED DESCRIPTION OF THE INVENTION

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 2 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 5 ("CPU"), a system memory 7, including a random access memory 9 ("RAM") and a read-only memory ("ROM") 11, and a system bus 12 that couples the memory to the CPU 5. 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 11. The computer 2 further includes a mass storage device 14 for storing an operating system 16, application programs, and other program modules, which will be described in greater detail below.

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 2. 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 2.

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 2.

According to various embodiments of the invention, the computer 2 may operate in a networked environment using logical connections to remote computers through a network 18, such as the Internet. The computer 2 may connect to the network 18 through a network interface unit 20 connected to the bus 12. It should be appreciated that the network interface unit 20 may also be utilized to connect to other types of networks and remote computer systems. The computer 2 may also include an input/output controller 22 for receiving and processing input from a number of other devices, including a keyboard, mouse, or electronic stylus (not shown in FIGURE 1). Similarly, an input/output controller 22 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 14 and RAM 9 of the computer 2, including an operating system 16 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 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 a Web browser application program 10. As known to those skilled in the art, the Web browser application program 10 is operative to request,

receive, render, and provide interactivity with electronic documents, such as a Web page 24 that has been formatted using HTML. Moreover, the Web browser application program 10 may be operative to execute scripts contained in the Web page 24, such as scripts expressed utilizing the JAVA SCRIPT language from SUN MICROSYSTEMS, INC. According to one embodiment of the invention, the Web browser application program 10 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 24 may comprise a viewable representation of a spreadsheet document. In particular, the Web page 24 may include HTML and scripts which, when displayed by the Web browser application 10, provide a visual display for a spreadsheet. Moreover, the scripts included in the Web page 24 allow a user of the computer to interact with the display provided by the Web browser application 10 and modify the spreadsheet. From the visual display provided by the Web browser application 10, the user may also be permitted to request that the native spreadsheet file be opened in a spreadsheet application program 26. In response to such a request, the spreadsheet application 26 will be launched and the native spreadsheet file corresponding to the spreadsheet expressed by the Web page 24 will be loaded by the spreadsheet application program 26. The spreadsheet application program 26 may comprise the EXCEL spreadsheet application program from MICROSOFT CORPORATION or another spreadsheet application program from another manufacturer. Additional details regarding the process for viewing and interacting with a spreadsheet from within the context of the Web browser application 10 will be provided below with respect to FIGURES 2-9.

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 2 is connected to a network 18. Also connected to the network 18 is an application server 30. The application server 30

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 30 is operative to execute a Web server application for receiving and responding to requests for documents stored at or accessible to the application server 30. Moreover, the application server 30 is operative to receive and respond to requests for pages generated by a Web application 34. It should be appreciated that the Web application 34 may comprise code executable at the application server 30, executable code for communicating with other computers, and may include templates, graphics, audio files, and other content known to those skilled in the art.

According to one aspect of the invention, the Web application 34 is operative to provide an interface to a user of the computer 2 to interact with a spreadsheet or workbook accessible via the network 18. In particular, the Web application 34 utilizes a server application programming interface ("API") 38. According to this embodiment of the invention, the server API 38 is operative to enable communication with a spreadsheet calculation server 32. The spreadsheet calculation server 32 is operative to execute a spreadsheet calculation server program 40. The spreadsheet calculation server program 40 comprises an executable program for retrieving and calculating spreadsheets, and such as the workbook 44 stored in the database 42. It should be appreciated that in the embodiments of the invention described herein, the spreadsheet calculation server program 40 comprises the EXCEL CALCULATION SERVER program from MICROSOFT CORPORATION. However, other programs for calculating a spreadsheet on a server computer may be utilized. It should also be appreciated that the calculation server 32 may include many of the conventional hardware and software components discussed above with respect to FIGURE 1.

As will be discussed in greater detail below, the computer 2 may transmit a request to the application server 30 to view the workbook 44 within the context of the Web browser application 10. In response to such a request, the Web application 34 communicates with the calculation server 32 through the server API 38. In particular, the Web application 34 requests from the calculation server 32 the

appropriate workbook 44. As will be discussed in greater detail below, in response to such a request, the calculation server program 40 retrieves the workbook 44 and converts the workbook 44 into the XMLSS format. As known to those skilled in the art, the XMLSS format comprises a markup language schema for expressing the contents of a spreadsheet. Once the spreadsheet calculation server program 40 has converted the requested workbook 44 to the XMLSS format, the XMLSS formatted file is returned to the Web application 34.

Once the Web application 34 has received from the calculation server 32 the XMLSS representation of the workbook 44, the application server 30 utilizes the spreadsheet Web renderer 36 to convert the XMLSS formatted spreadsheet into a representation of the spreadsheet that may be rendered by the Web browser application 10. In particular, the spreadsheet Web renderer 36 converts an XMLSS formatted document into a document containing HTML that may be rendered by the Web browser application 10 to display the spreadsheet. Moreover, according to embodiments of the invention, the spreadsheet Web renderer 36 is capable of outputting HTML which, when rendered by the Web browser application 10, appears substantially similar to the output of the spreadsheet application 26 when utilized to view the same workbook 44. Once the spreadsheet Web renderer 36 has completed rendering the file, it is returned by the application server 30 to the computer 2 where it may be rendered by the Web browser application 10.

As will also be discussed in greater detail below, the spreadsheet Web renderer 36 may also be operative to render into the markup language file one or more scripts for allowing the user of the computer 2 to interact with the spreadsheet within the context of the Web browser application 10. In order to provide this functionality, the spreadsheet Web renderer 36 is operative to render script code that is executable by the Web browser application 10 into the returned Web page. The scripts may provide functionality, for instance, for allowing a user to view a larger portion of the spreadsheet, to modify pivot tables contained within the spreadsheet, to load the native version of the spreadsheet in the spreadsheet application 26, and to perform other functions.

In order to provide interactivity with the spreadsheet within the context of the Web browser application 10, the Web browser application 10 receives user input. In response to certain types of user input, the scripts may be executed. When a script is executed, a response may be transmitted to the application server 30 indicating an element within the spreadsheet that has been acted upon, to identify the type of interaction that was made, and to further identify to the Web application 34 the function that should be performed upon the appropriate element. In response to receiving such a request, the Web application may make additional requests to the calculation server 32 for an updated workbook 44. In response thereto, the spreadsheet calculation server program 40 may recalculate the workbook 44 in view of the user action and provide an XMLSS formatted representation of the updated workbook to the Web application 34. The spreadsheet Web renderer 36 is then operative to render the updated workbook into a format renderable by the Web browser application 10. The Web application 34 is then operative to transmit the updated spreadsheet to the computer 2 where it may be rendered by the Web browser application 10. This process may be repeated any number of times as the user interacts with the spreadsheet from within the context of the Web browser application 10. When a user is finished editing the spreadsheet within the Web browser application 10, any changes made to the spreadsheet may be propagated through the Web application 34, the calculation server 32, and to the workbook 44 stored in the database 42. Additional details regarding these various processes will be provided below with respect to FIGURES 3-9.

Referring now to FIGURE 3, an illustrative routine 300 will be described illustrating a process for viewing and interacting with a spreadsheet from within the context of a Web browser application program. It should be appreciated that although the embodiments of the invention described herein are presented in the context of a Web browser application program, the invention may be utilized in other types of application programs that support the rendering of markup language documents. For instance, the embodiments of the invention described herein may be utilized within a personal information manager 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.

When reading the discussion of the routines 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 FIGURES 3 and 6, and making up 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.

The routine 300 begins at operation 302, where the client computer 2 transmits to the application server 30 a request for the workbook 44. The request may be transmitted by the computer 2 in response to a user requesting through the Web browser application 10 that the workbook 44 be opened for viewing and editing within the Web browser application 10. It should be appreciated that the application server 30 may provide a Web based interface 30 for allowing a user to view the available workbooks and to select one of the workbooks for viewing and editing.

From operation 302, the routine 300 continues to operation 304, where the application server 30 transmits a request to the calculation server 32 for the workbook 44. In particular, the Web application 34 may utilize the server API 38 to communicate with the spreadsheet calculation server program 40. Through this communication path, a request may be made for the workbook 44.

From operation 304, the routine 300 continues to operation 306, where the spreadsheet calculation server program 40 retrieves the requested workbook 44 from the database 42. The spreadsheet calculation server program 40 then calculates the

contents of the workbook 44 at operation 308. The calculation operation 308 may include calculations similar to those performed by a spreadsheet application 26 when recalculating the workbook 44.

Once the calculation server 32 has calculated the workbook 44, the spreadsheet calculation server program 40 is also operative to convert the calculated workbook to the XMLSS format. As discussed above, the XMLSS format comprises a markup language schema for describing the contents of a spreadsheet. It should be appreciated that, according to aspects of the invention, only a portion of the workbook 44 is converted to the XMLSS format. This allows only a displayable portion of the spreadsheet to be returned to the computer 2. Additional portions may be requested through the interface provided by the Web page 24. Alternatively, the entire workbook 44 may be converted to the XMLSS format and returned to the application server 30.

At operation 312, the calculation server 32 responds to the request from the application server 30 with the XMLSS formatted workbook. The routine 300 then continues to operation 314, where the spreadsheet Web renderer 36 executing on the application server 30 converts the XMLSS formatted workbook to a format that can be rendered and displayed by the Web browser application 10. In particular, the spreadsheet Web renderer 36 converts the XMLSS workbook to a Web page 24 that includes both HTML and scripts. The HTML is utilized by the Web browser 10 to display the contents of the requested workbook 44. The scripts are utilized by the Web browser application 10 to allow a user to interact with the rendered version of the spreadsheet within the context of the Web browser application 10.

From operation 314, the routine 300 continues to operation 316, where the application server responds to the original request from the computer 2 for the workbook 44. In particular, the application server 30 transmits the Web page 24 to the computer 2 that includes the HTML and scripts necessary to view and interact with the workbook 44 within the context of the Web browser application 10.

Once the computer 2 has received the Web page 24, the Web browser application 10 is operative to render the contents of the Web page 24. In particular, the HTML is rendered in order to provide a view of the workbook 44 that is substantially

similar to that which would be provided by the spreadsheet application 26 if the same workbook 44 were opened utilizing the spreadsheet application 26. Moreover, the Web browser application 10 is operative to render the scripts so that a user of the computer 2 may interact with the spreadsheet within the context of the Web browser application 10. Additional details regarding the interaction with the spreadsheet by a user will be provided below with respect to FIGURES 6-9.

Referring now to FIGURE 4, an illustrative screen display provided by a spreadsheet application program 26 when utilized to open a workbook 44 for viewing and editing will be described. In particular, FIGURE 4 shows a user interface window 400 provided by a spreadsheet application program 26 when utilized to open a workbook 44 for viewing and editing. The window 400 includes a number of conventional controls, including a command menu 50, a toolbar 52, and a formatting toolbar 56. As known to those skilled in the art, the menus and icons provided by these user interface toolbars may be utilized by a user of the spreadsheet application 26 to access the various features of the spreadsheet application program 26.

As also shown in FIGURE 4, the window 400 includes a grid 58 having one or more rows and one or more columns. The columns and rows are familiar to users of conventional spreadsheet application programs and define the workspace for the currently selected workbook. As also known to those skilled in the art, each of the cells defined by a particular row and column combination can be edited with numerical or text data and commands may be performed on the various cells to perform virtually unlimited types of analysis operations. The various operations and functions available through the spreadsheet application program 26 are well known to those skilled in the art.

The window 400 also includes various user interface elements for navigating within the currently selected spreadsheet or within other spreadsheets within the currently selected workbook. For instance, the window 400 shown in FIGURE 4 includes user interface buttons 54A-54C which, when selected by a user, cause the appropriate spreadsheet to be displayed within the window 400. The other user interface items within the window 400 for navigating within the current spreadsheet or

to other spreadsheets, such as the user interface slider bars shown in FIGURE 4, are well known to those skilled in the art. As will be described in greater detail below, the various embodiments of the invention provided herein allow the Web browser application 10 to provide a visual display that is substantially similar to the display provided by the spreadsheet application 26 when viewing the same workbook 44. A screen display provided by the Web browser application 10 when utilized to view the same workbook as utilized by the spreadsheet application 26 to generate the window 400 shown in FIGURE 4 will be described below with reference to FIGURE 5.

As discussed briefly above, FIGURE 5 illustrates the screen display provided by the Web browser application 10 when utilized to open a workbook 44. In particular, the screen display illustrated in FIGURE 5 illustrates the screen display provided by the Web browser application 10 when utilizing the various aspects of the present invention to open the same workbook 44 as opened by the spreadsheet application 26 and utilized to provide the screen display shown in FIGURE 4. It should be appreciated that the screen display shown in FIGURE 5 provided by the Web browser application 10 is substantially similar to the display provided by the spreadsheet application 26 shown in FIGURE 4.

As shown in FIGURE 5, the window 500 provided by the Web browser application 10 includes a number of conventional components associated with Web browser application programs. In particular, the window includes a menu bar 60, a tool bar 62, and an address bar 64. As known to those skilled in the art, these user interface items provide access to the various functions provided by the Web browser application 10. When utilizing the various aspects of the present invention described herein, the Web browser application 10 is also operative to provide a screen display that includes a grid 68. It should be appreciated that the grid 68 shown in FIGURE 5 is substantially similar to the grid 58 shown in FIGURE 4. In particular, the grid 68 shown in FIGURE 5 defines a number of rows, columns, and corresponding cells. Moreover, the cells shown in the grid 68 include the same content as the cells shown in the grid 58. The cells of the grid 68 are also formatted utilizing the same font, color, justification, and other attributes as the contents of the cells shown in the grid 58 in FIGURE 4.

As also shown in FIGURE 5, the display provided by the Web browser application 10 includes a number of user interface buttons within the rendered page for performing various functions on the currently shown workbook. For instance, the button 66A may be selected in order to open the currently selected workbook within the spreadsheet application 26. The user interface button 66B may be utilized to save any modifications made to the currently opened workbook to the version of the workbook 44 stored at the calculation server 32. Additionally, a button 66C may be utilized to print the contents of the currently opened workbook.

Other buttons may be utilized to navigate between several spreadsheets contained within a single workbook. For instance, the user interface button 70A-70C shown within the context of the Web browser application 10 in FIGURE 5 perform a similar function to the buttons 54A-54C illustrated in FIGURE 4 with respect to the display provided by the spreadsheet application program 26. Additional details will be provided below with respect to FIGURES 6-9 regarding the interactivity provided by the embodiments of the invention within the Web browser application 10.

Turning now to FIGURE 6, an illustrative routine 600 will be described that illustrates various aspects of the invention for allowing a user to interact with a workbook rendered within the context of the Web browser application 10. The routine 600 begins at operation 602, where the contents of the Web page 24, including the HTML necessary for displaying the Web page and the scripts necessary for allowing a user to interact with the spreadsheet are rendered. The routine 600 then continues to operation 604, where the Web browser application 10 receives user input for interacting with the displayed spreadsheet. For instance, the Web browser application 10 may receive from the user input modifying the contents of a pivot table, requesting that another spreadsheet in a workbook be displayed, for saving any modifications to the spreadsheet, or for performing other functions. It should be appreciated that one or more scripts may be executed by the Web browser application 10 to receive the input and perform the communication with the application server 30 necessary to cause the interaction to be performed.

Once input has been received from a user at operation 604, the appropriate script transmits data to the application server 30 identifying the particular action taken by the user at operation 606. At operation 608, the application server 30 determines the particular action taken by the user. Once the requested action has been identified, the application server 30 transmits to the calculation server 32 a command requesting that the particular action be taken on the specific workbook. As discussed above, the application server 30 communicates with the calculation server 32 through the server API 38.

In response to receiving the request from the application server 30 to perform a command on the current workbook, the spreadsheet calculation server program 40 performs the requested action on the workbook 44. For instance, if a user has requested that elements within a pivot table be rearranged, the spreadsheet calculation server 40 performs the actual rearrangement of the elements. Once the action has been performed, the spreadsheet calculation server program 40 recalculates the workbook at operation 612. The recalculated workbook reflects the changes made by the user to the currently open workbook through the Web browser application 10.

Once the workbook 44 has been recalculated, the calculation server 32 converts the recalculated workbook to the XMLSS format. This process is described above with reference to FIGURE 3. Once the recalculated workbook has been converted to the XMLSS format, the recalculated workbook is transmitted from the calculation server 32 to the application server 30. At the application server 30, the spreadsheet Web renderer 36 converts the XMLSS version of the recalculated spreadsheet to a format renderable by the Web browser application 10. The renderable version of the recalculated spreadsheet is then transmitted to the Web browser application 10 where it may be rendered and displayed to a user. This process is described above with reference to FIGURE 3 and operations 312-318.

Referring now to FIGURES 7-9, additional details will be provided regarding a user's ability to interact with a spreadsheet within the context of the Web browser application 10. In particular, as shown in FIGURE 7, a window 700 generated by the Web browser application 10 is shown. The window 700 includes a pivot table 72

as part of a spreadsheet displayed within the Web browser application 10 in the manner described above. The pivot table 72 includes a user interface button 74 for causing a dropdown menu to be displayed. Through the dropdown menu, a user or group of users may be selected. In response to the selection of such a user, the pivot table 72 will be updated to reflect this selection.

As an example, in FIGURE 8, the button 74 has been selected to cause the dropdown menu 76 to be displayed by the Web browser application 10. The user may then select from the dropdown menu 76 one of the displayed names. As an example, if the user selects the name "Blythe", the contents of the pivot table 72 will be recalculated and redisplayed as shown in FIGURE 9.

In order to recalculate and redisplay the contents of the pivot table 72 in response to the selection of the name from the dropdown menu 76, a request must be sent from the client computer to the application server 30 and processed by the calculation server 32 as described above with reference to FIGURE 6. Once the workbook has been recalculated, it may be transmitted back to the Web browser application program 10 as described above with respect to FIGURE 3. In this manner, virtually any type of interaction may be performed on the spreadsheet displayed within the Web browser application program 10.

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 enabling a spreadsheet to be viewed and modified within the context of a Web browser application program. 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.

WE CLAIM:

1. A method for viewing and interacting with a spreadsheet from within the context of a Web browser application program, the method comprising:

receiving a request to open a spreadsheet document in the Web browser application program;

in response to the request, generating a representation of the spreadsheet document capable of being rendered by the Web browser application program;

transmitting the representation of the spreadsheet document to the Web browser application program; and

providing a viewable version of the spreadsheet document by rendering the representation of the spreadsheet document in the Web browser.

2. The method of Claim 1, wherein the representation of the spreadsheet document comprises only markup language tags for use in rendering the representation of the spreadsheet document and scripts executable within the context of the Web browser for providing interactivity with the viewable version of the spreadsheet.

3. The method of Claim 2, wherein the viewable version of the spreadsheet appears substantially similar to a visual display provided by a spreadsheet application program when utilized to view the spreadsheet document.

4. The method of Claim 3, further comprising:

receiving a user request to modify an aspect of the spreadsheet rendered within the Web browser; and

in response to the request, executing a one of the scripts to perform the requested modification.

5. The method of Claim 4, wherein executing a one of the scripts to perform the requested modification comprises transmitting to a server computer data identifying the requested modification, and wherein the method further comprises:

receiving the data identifying the requested modification;

in response to receipt of the data, generating a recalculated spreadsheet document, the recalculated spreadsheet document reflecting the requested modification;

generating a representation of the recalculated spreadsheet document capable of being rendered by the Web browser application program; and

transmitting the representation of the recalculated spreadsheet document to the Web browser application program.

6. The method of Claim 5, wherein generating a representation of the spreadsheet document capable of being rendered by the Web browser application program comprises:

retrieving the spreadsheet document;

converting the spreadsheet document into a document expressed using the extensible markup language spreadsheet format; and

converting the document expressed using the extensible markup language spreadsheet format into a representation of the spreadsheet document expressed using a hypertext markup language and including markup language tags and scripts capable of being rendered by the Web browser application.

7. A computer-readable medium having computer-executable instructions stored thereon which, when executed by a computer, will cause the computer to perform the method of Claim 1.

8. A computer-controlled apparatus capable of performing the method of Claim 1.

9. A method for viewing and interacting with a spreadsheet from within a Web browser application program, the method comprising:

receiving a request for a workbook comprising one or more spreadsheets, the request generated from within the Web browser application program;

in response to the request, retrieving the requested workbook, calculating the requested workbook, and converting the calculated workbook to an extensible markup language spreadsheet format;

converting the calculated workbook from the extensible markup language spreadsheet format into a renderable document, the renderable document comprising hypertext markup language tags and scripts capable of being rendered by the Web browser application; and

responding to the request for a workbook by providing the renderable document.

10. The method of Claim 9, wherein the renderable document comprises only markup language tags for use in rendering the representation of the spreadsheet document and scripts executable within the context of the Web browser for providing interactivity with the viewable version of the spreadsheet.

11. The method of Claim 10, wherein the renderable document appears substantially similar to a visual display provided by a spreadsheet application program when utilized to view the requested workbook.

12. The method of Claim 11, further comprising:

receiving a request to modify the workbook, the request including data identifying a modification to be made to the workbook; and

in response to the request, generating a recalculated workbook, the recalculated workbook reflecting the requested modification,

generating a renderable document for the recalculated workbook, the renderable document capable of being rendered by the Web browser application program, and

transmitting the renderable document for the recalculated workbook to the Web browser application program.

13. A computer-readable medium having computer-executable instructions stored thereon which, when executed by a computer, will cause the computer to perform the method of Claim 9.

14. A computer-controlled apparatus capable of performing the method of Claim 9.

15. A system for allowing a spreadsheet to be viewed and modified from within the context of a Web browser application program, the system comprising:

a first server computer operative to execute program code for receiving a request for a renderable version of a spreadsheet from the Web browser application program, for requesting from a second server computer a version of the spreadsheet expressed using the extensible markup language spreadsheet (XMLSS) format, for receiving from the second server computer the XMLSS version of the spreadsheet, for converting the XMLSS version of the spreadsheet to a renderable version of the spreadsheet, and for responding to the request from the Web browser with the renderable version of the spreadsheet; and

a second server computer operative to receive the request from the first server computer for the XMLSS version of the spreadsheet, to retrieve a native version of the spreadsheet from a database, to convert the native version of the spreadsheet to the XMLSS version of the spreadsheet, and to respond to the request from the first server computer with the XMLSS version of the spreadsheet.

16. The system of Claim 15, wherein the renderable version of the spreadsheet document comprises only markup language tags for use in rendering the spreadsheet document and scripts executable within the Web browser for providing interactivity with the viewable version of the spreadsheet.

17. The system of Claim 16, wherein the renderable version of the spreadsheet appears substantially similar to a visual display provided by a spreadsheet application program when utilized to view the native version of the spreadsheet.

18. The system of Claim 17, further comprising a Web browser application program operative to request from the first server computer the renderable version of the spreadsheet and further operative to render the renderable version of the spreadsheet when received from the first server computer.