



(19) **United States**

(12) **Patent Application Publication**  
**Cooper et al.**

(10) **Pub. No.: US 2015/0199308 A1**

(43) **Pub. Date: Jul. 16, 2015**

(54) **SYSTEMS AND METHODS FOR CONTROLLING THE DISPLAY OF ONLINE DOCUMENTS**

(52) **U.S. CL.**  
CPC ..... **G06F 17/2247** (2013.01); **H04L 65/403** (2013.01)

(75) Inventors: **Sarah Cooper**, New York, NY (US);  
**Mikhail Leder**, Brooklyn, NY (US)

(57) **ABSTRACT**

(73) Assignee: **Google Inc.**, Mountain View, CA (US)

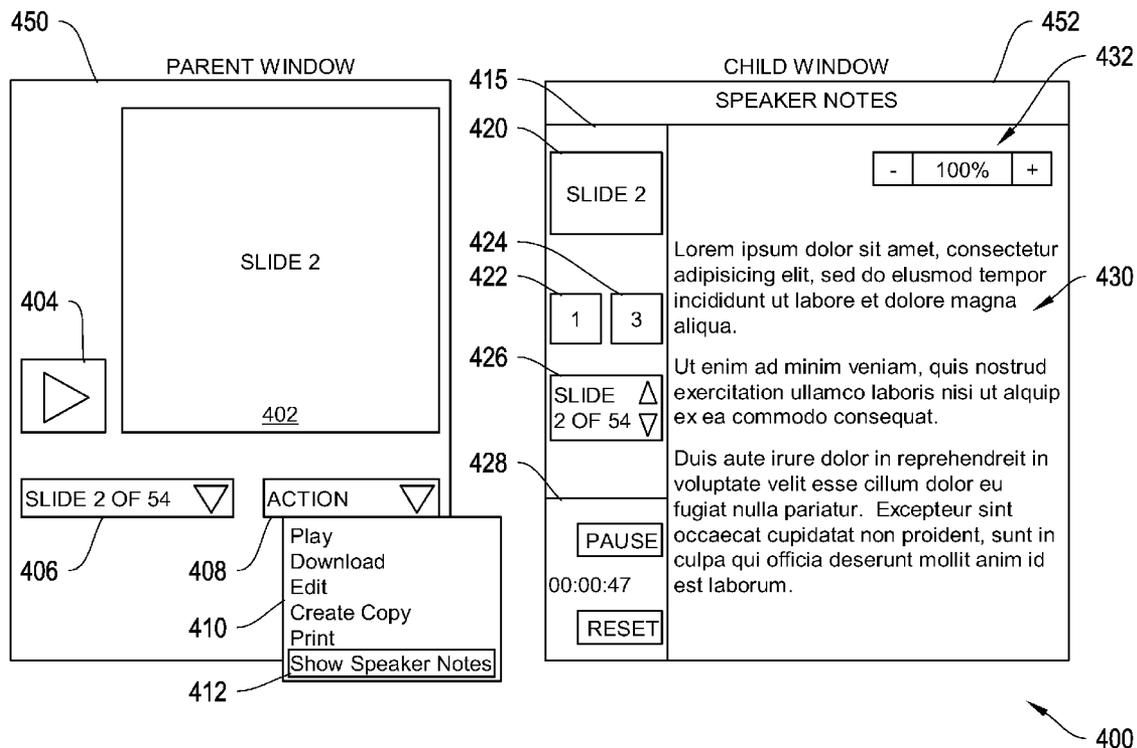
(21) Appl. No.: **13/275,123**

(22) Filed: **Oct. 17, 2011**

In many aspects, the systems and methods described herein include software and hardware for controlling, from one browser window, the display of information in an online document processing application on another browser window. For example, the systems and methods described herein provide a user with the ability to view a presentation slide deck on a first browser window while simultaneously viewing speaker notes for each slide of the slide deck in a second browser window. The systems and methods not only provide the user with the ability to view speaker notes, but also control the slideshow from the second browser window.

**Publication Classification**

(51) **Int. Cl.**  
**G06F 17/22** (2006.01)  
**H04L 29/06** (2006.01)  
**G06F 3/048** (2006.01)



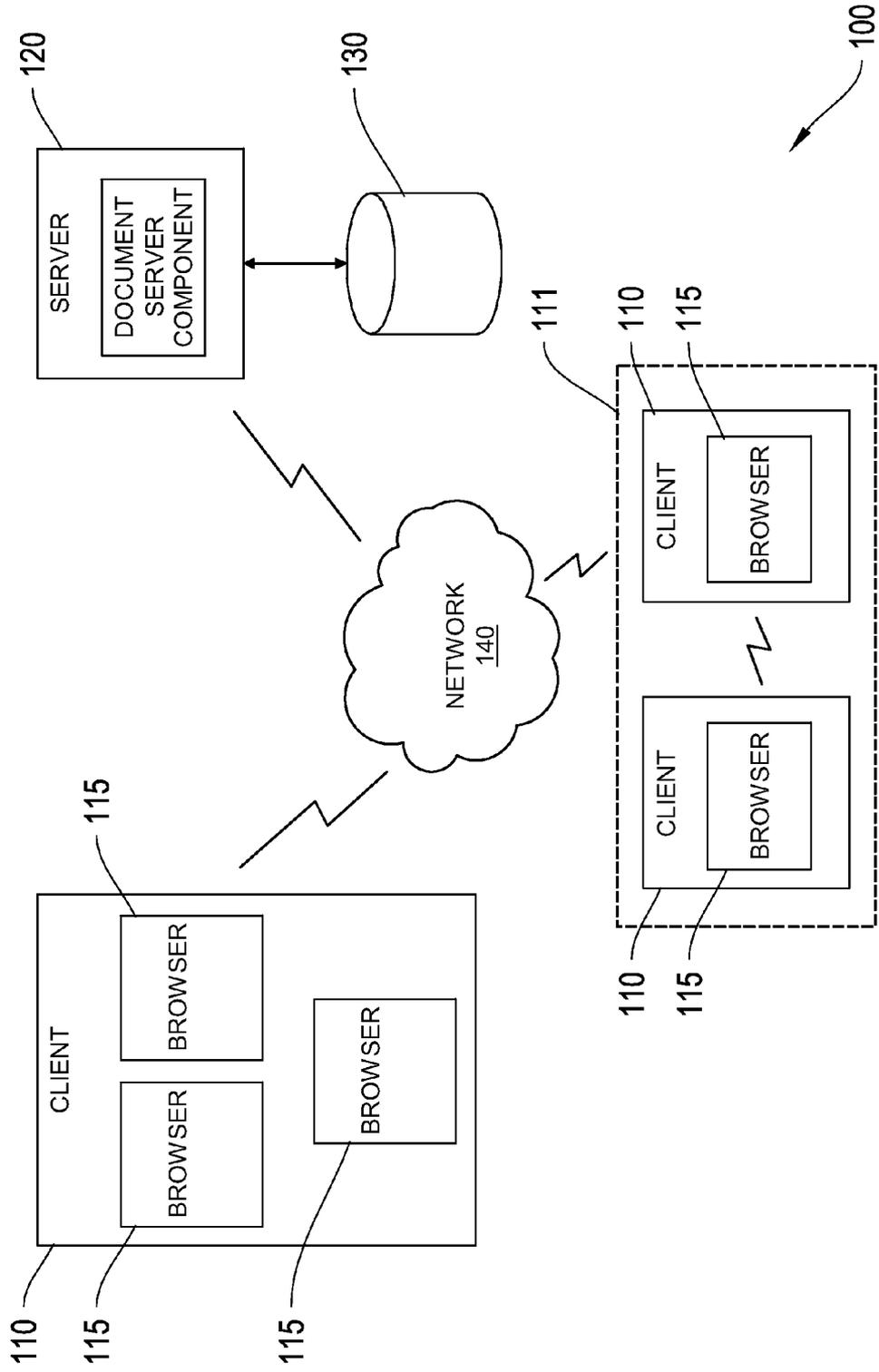


FIG. 1

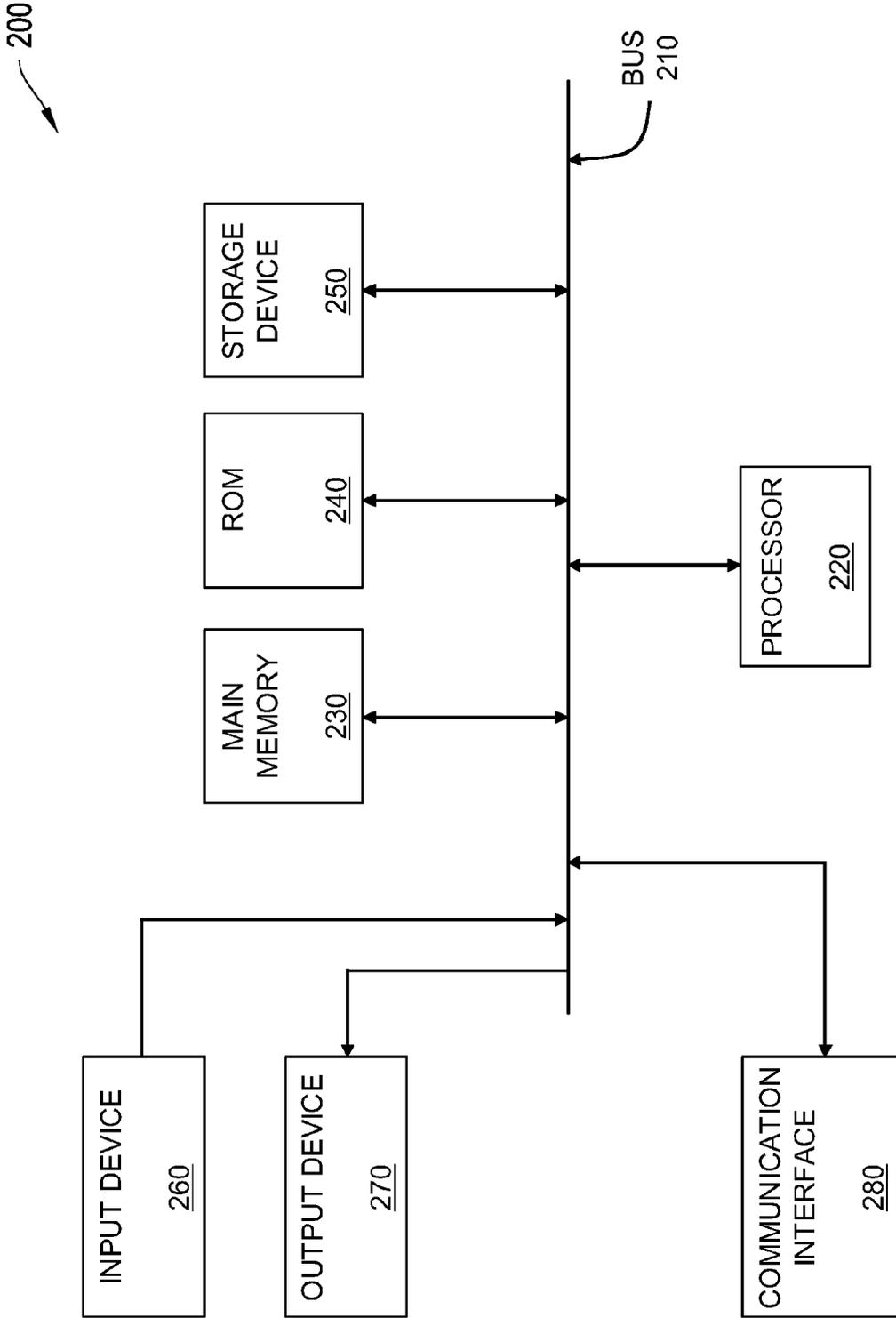


FIG. 2

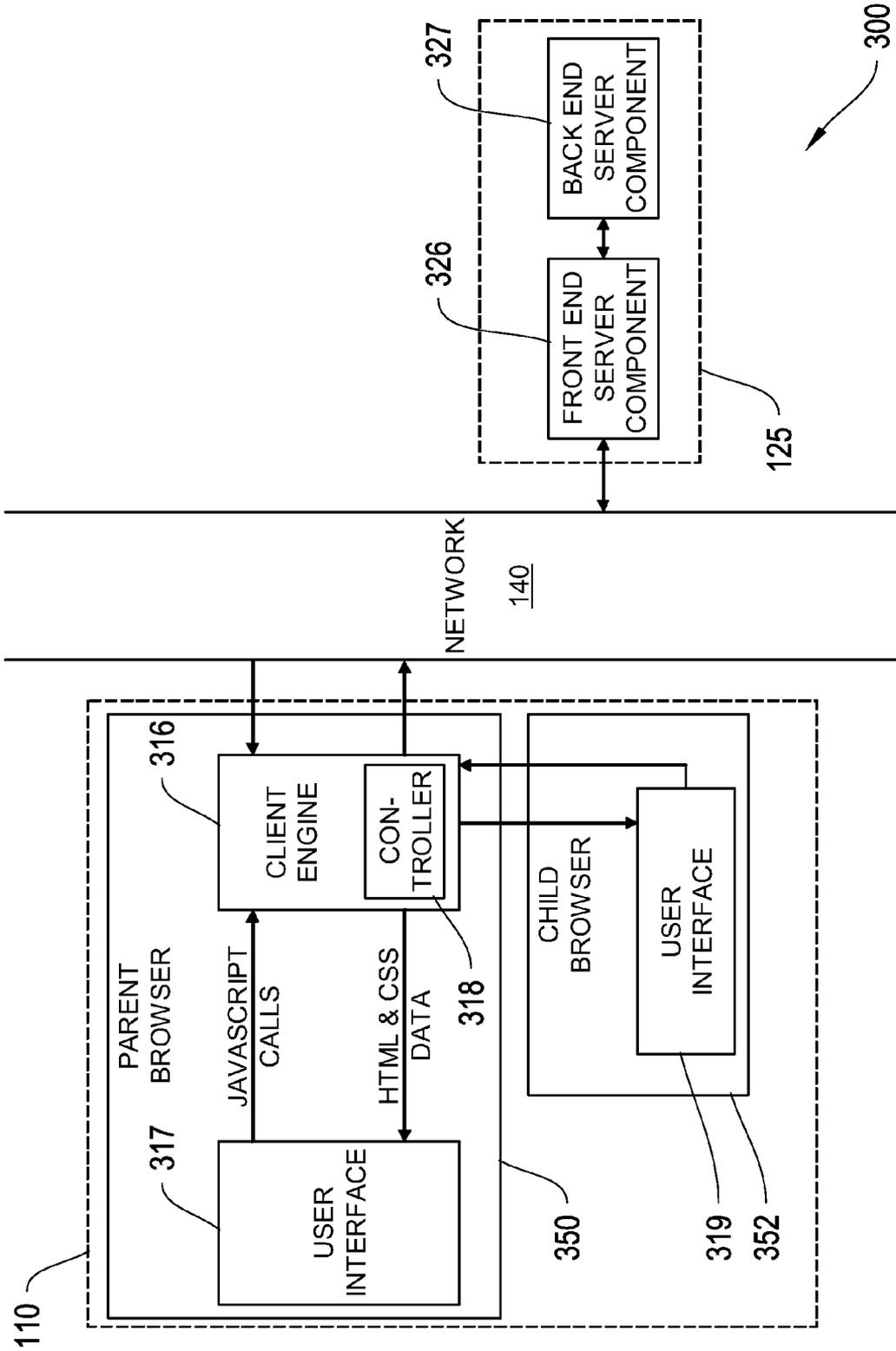


FIG. 3

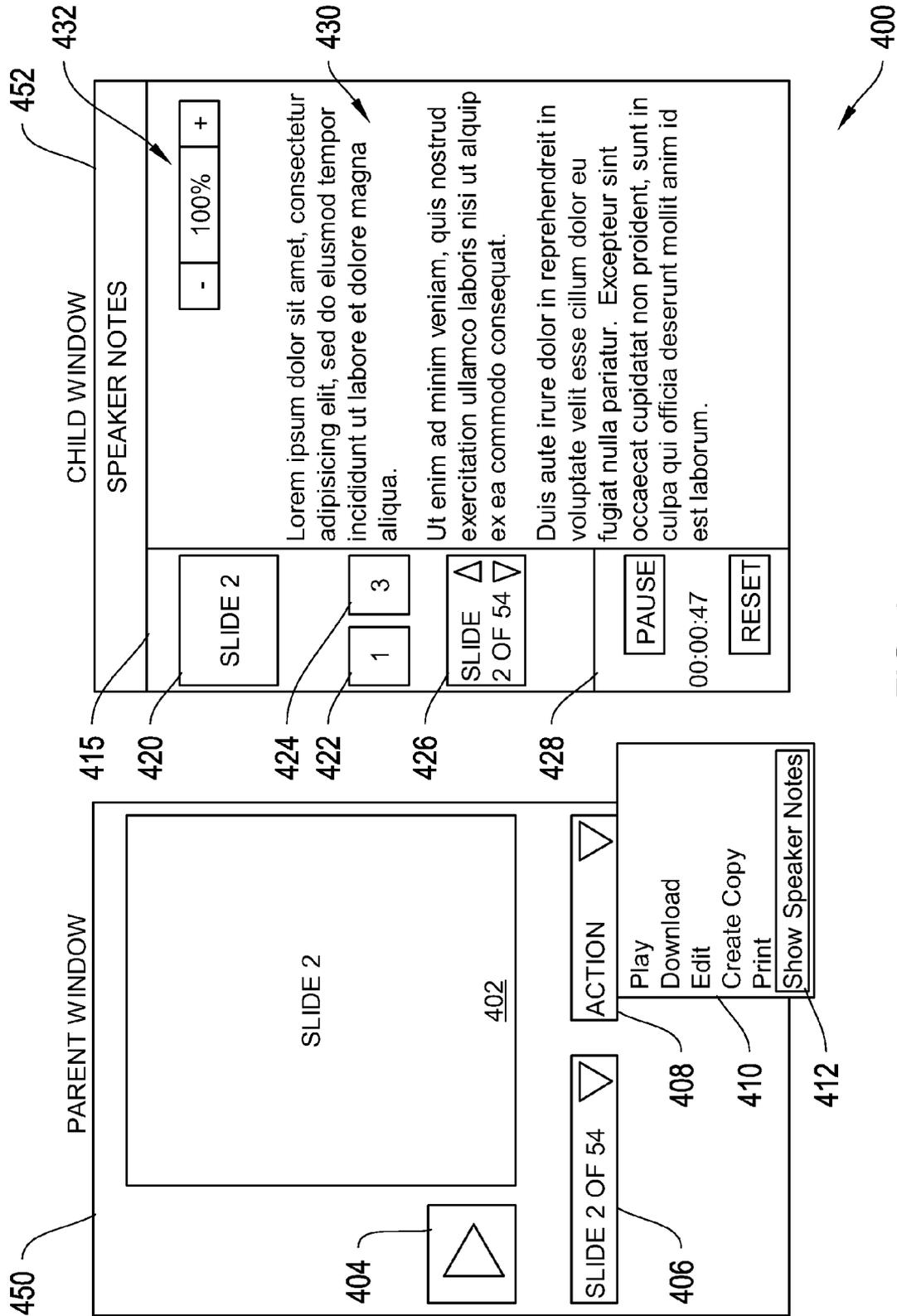


FIG. 4

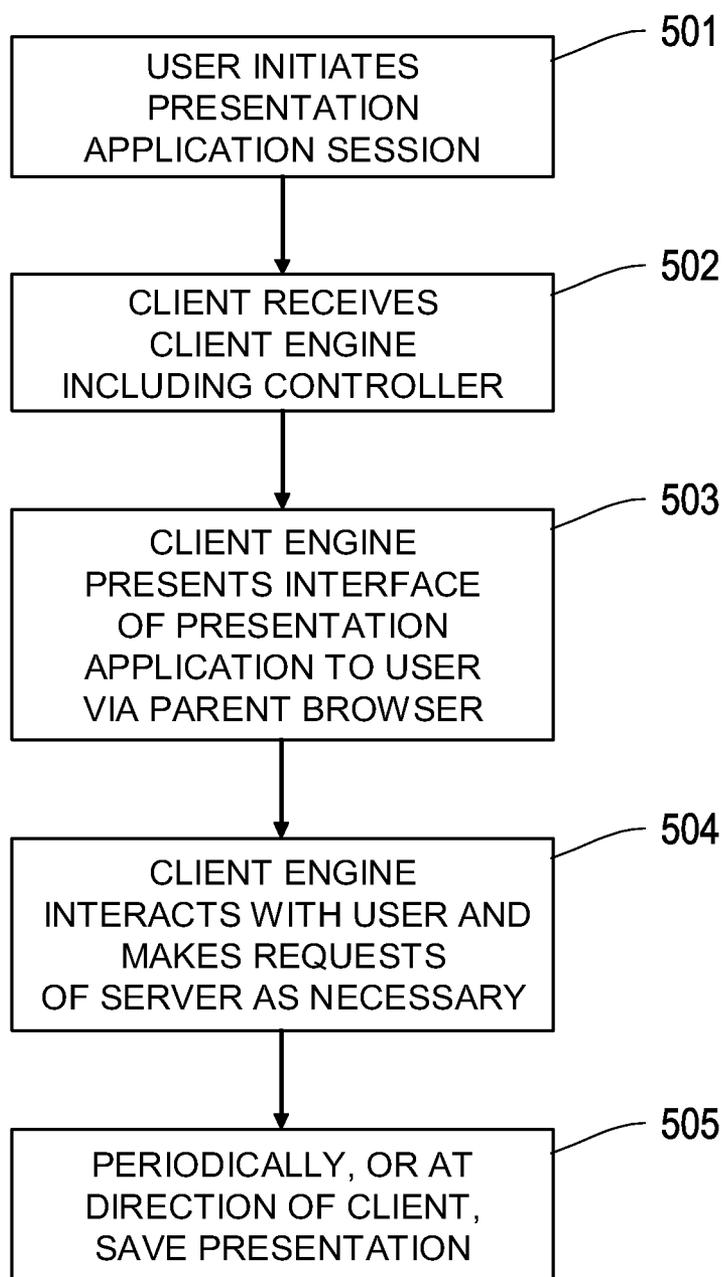


FIG. 5

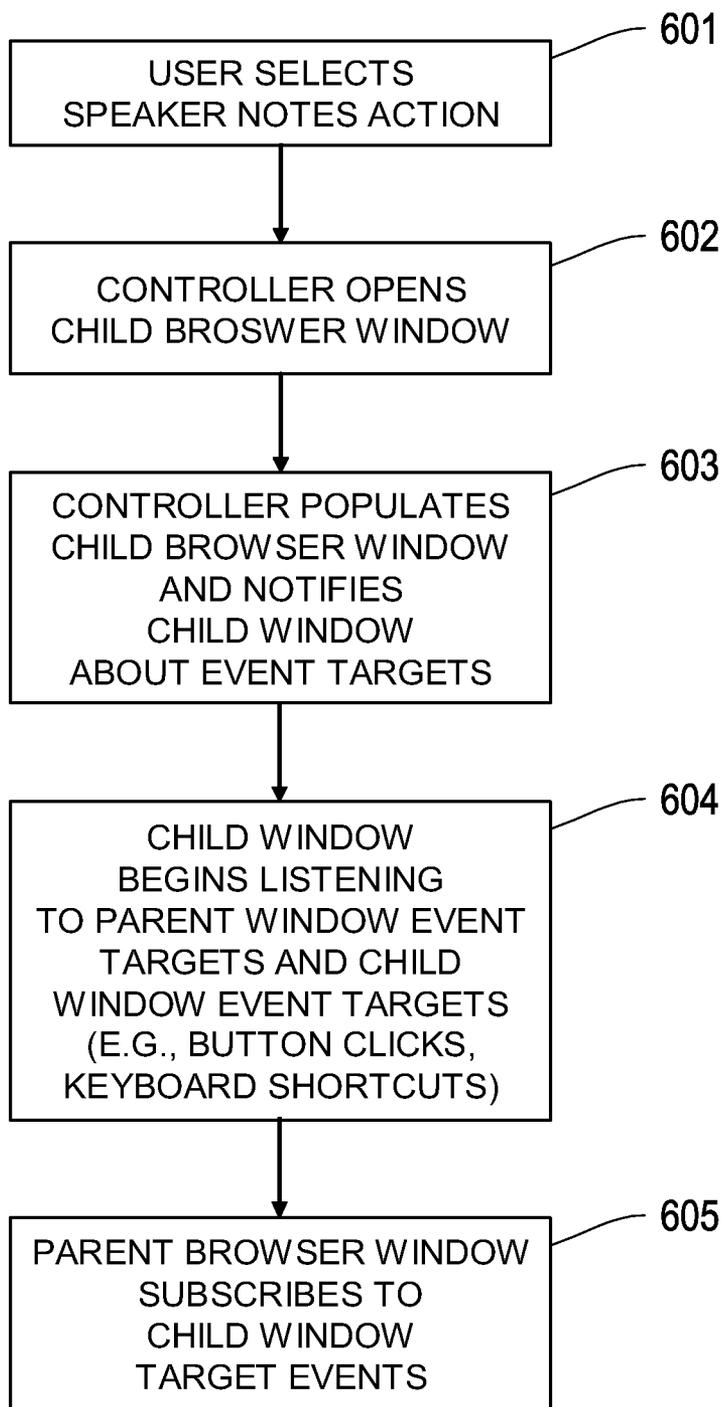


FIG. 6

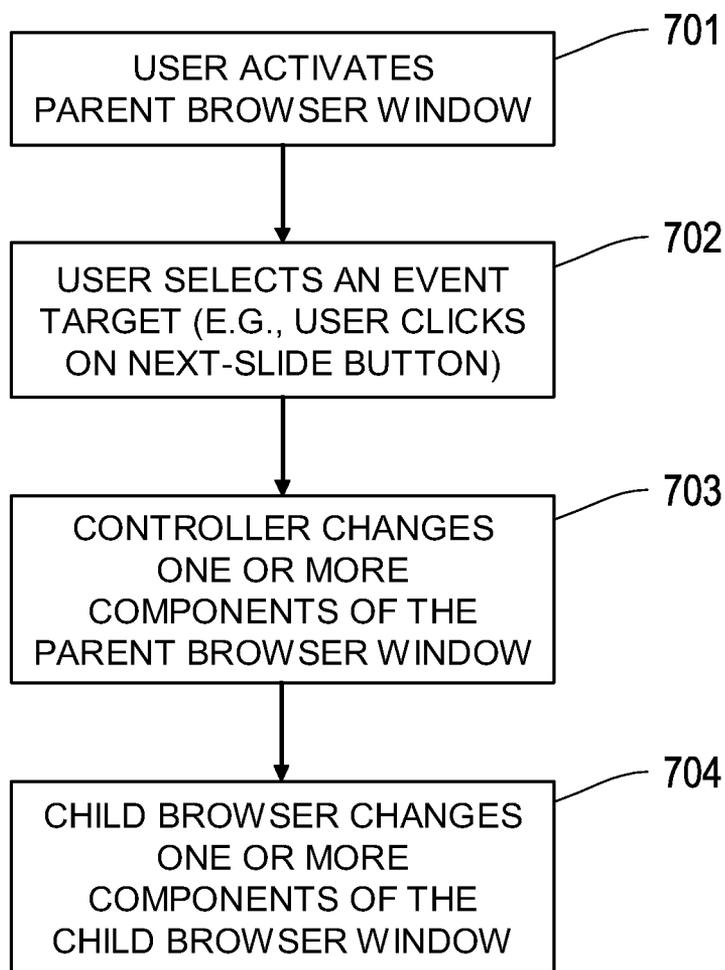


FIG. 7

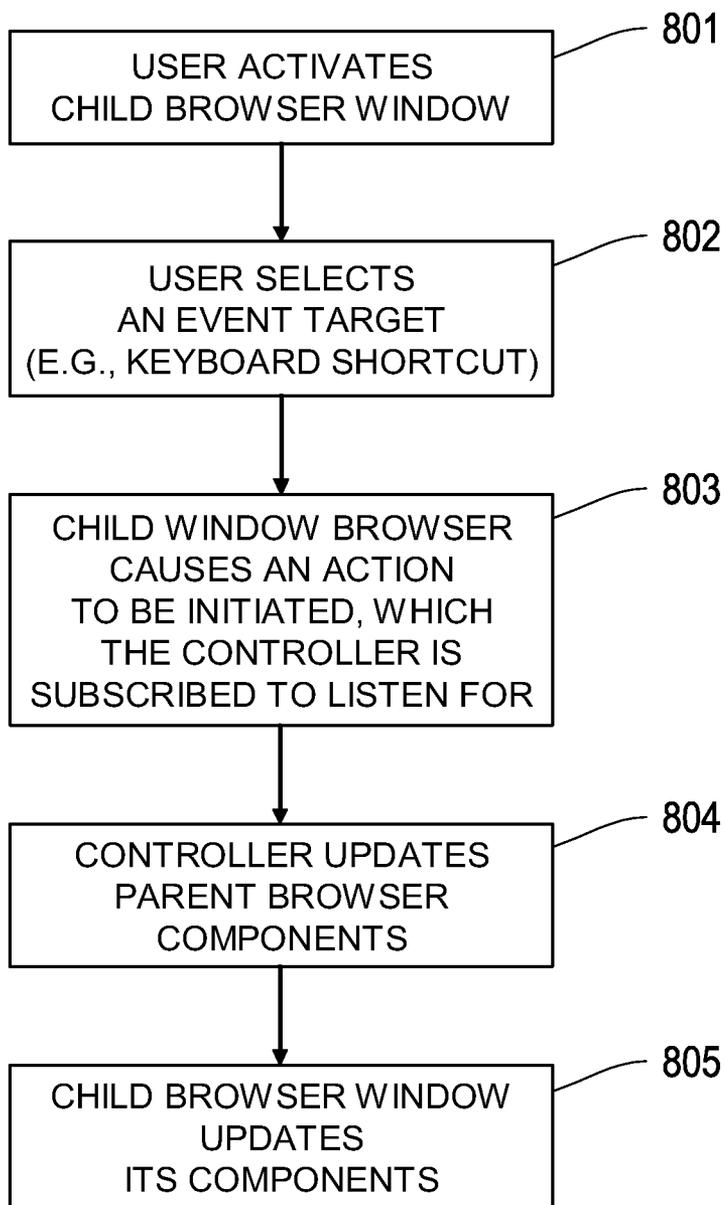


FIG. 8

**SYSTEMS AND METHODS FOR CONTROLLING THE DISPLAY OF ONLINE DOCUMENTS**

**BACKGROUND**

**[0001]** Electronic document processing applications are in widespread use in various industries including finance, business, law and academia. One type of electronic document processing application is a slideware application or presentation application. A presentation application is typically used to produce a presentation that is often a collection of slides or documents having media including text, pictures, embedded audio and video, and other rich media and interactive content.

**[0002]** Existing presentation applications tend to be software programs that execute locally on a computer system of a single user. The presentation slides produced by such a system can be difficult to distribute and difficult to modify or collaborate on when working in a group environment.

**[0003]** Recently, online presentation applications have been developed which have the ability for a number of users, potentially at physically remote locations relative to one another, to easily collaborate and modify presentation slides. Additionally, the online presentation application may allow users to present their slides at any remote location having online access.

**[0004]** However, presentation produced by online presentation applications lack some of the functionality of offline software-based presentation applications.

**SUMMARY**

**[0005]** A feature missing from current online document processing applications, especially online presentation applications, is the ability to control, from one browser window, the presentation of an online document running in another browser window. Such a feature is especially useful in the context of using presentation applications to present a slide deck. When presenting, users want to be able to do two things: control the progression of slides in the presentation and review any notes (or speaker notes) that they may have prepared to assist them with their presentation. It is also desirable that the speaker notes not appear on the same window as the slides. Current-day online presentation applications allow users to view their presentation slides in one browser window and their speaker notes in another separate browser window. Thus, the users can view their speaker notes, while their audience views, separately, the presentation. However, these online presentation applications do not allow the users to control the progression of slides from the speaker notes window. Therefore, users would have to jump back and forth between the two windows to progress the slides and review their speaker notes. The present disclosure describes systems and methods that provide this missing feature.

**[0006]** Generally, the systems and methods described herein are directed to computerized methods for controlling from one browser window, an online application being displayed on another browser window. More particularly, in one aspect, the systems and methods described herein are directed to computerized methods for rendering and controlling a presentation document in an online presentation application. The methods may include displaying, on a first browser window at a client computing device, a first portion of a presentation document. The methods may further include receiving, in a second browser window at the client computing device, a

request for displaying a second portion of the presentation document on the first browser. The methods may then include replacing, in the first browser window of the client computing device, the displayed first portion of the presentation document with the second portion of the presentation document based on the request received at the second browser window. In certain embodiments, the first portion and second portion of the presentation document includes one or more presentation slides.

**[0007]** The methods may include opening the second browser window in response to a request received at the first browser window. The methods may further include displaying, on the second browser window at the client computing device, one or more user input components for allowing users to submit a request. The one or more user input components may include at least one of an image, button, text box, radio-button, and check-box. The methods described herein may further include displaying, on the second browser window at the client computing device, a third portion of the presentation document, which may include one or more pages of notes. The first browser window may receive a request for displaying the third portion of the presentation document in this second browser window.

**[0008]** In certain embodiments, the first portion of the presentation document is replaced with the second portion of the presentation document in the first browser window based on the request received at the first browser window. The first browser window may be configured to display speaker notes and the computerized methods may further include displaying, on the first browser window, a third portion of the presentation document including one or more pages of notes.

**[0009]** In another aspect, the systems and methods described herein include computerized methods for rendering and controlling Hypertext Markup Language (HTML) data and Cascading Style Sheets (CSS) data in a browser. The methods may include receiving, in a first browser window at a client computing device, a first set of data including HTML, CSS and Javascript data, and receiving, in the second browser window at the client computing device, a request for replacing the first set of data with a second set of data including HTML, CSS and Javascript data. The methods may further include replacing, in the first browser window of the client computing device, the first set of data with the second set of data based on the request received in the second browser window. The first browser window may be configured to listen for requests received in the second browser window. The methods may further include displaying, on the second browser window at the client computing device, one or more user input components for allowing users to submit a request. The second browser window may also be configured to receive a third set of data including HTML and CSS data.

**[0010]** In certain embodiments, the first browser window receives the second set of data from a server and creates the second browser window. The first browser window may also receive requests submitted by users. The first set of data may be replaced with the second set of data based on such a request in the first browser window.

**[0011]** In another aspect, the systems and methods described herein include systems for rendering and controlling a presentation document in an online presentation application. The systems may include a first browser window and a second browser window. The first browser window may include a client engine for running the online presentation application, and a user interface for displaying a portion of the

presentation document to a user and receiving requests from the user. The second browser window may include a user interface for controlling the presentation document displayed on the first browser window. The second browser may also include a user interface for receiving a portion of the presentation document from the user. In certain embodiments, the client engine may include a controller for generating the second browser window, listening to requests received at the second browser window, and replacing the displayed portion of the presentation document in the first browser with another portion of the presentation document based on the received requests.

[0012] According to the system, the client engine may be configured to provide HTML, CSS and Javascript data to the user interface in the first browser. The client engine may be configured to receive Javascript calls from the user interface in the first browser. In certain embodiments, the client engine is configured to receive data corresponding to the presentation document and online presentation application data from a server. The user interface in at least one of the first browser and the second browser may include components receptive to inputs received from at least one of a mouse, keyboard, HTML frame/object, and HTML form. The components may be receptive to at least one of mouse double-clicks, mouse down, mouse up, mouse over, mouse move, mouse out, key down, key press, and key up.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The above and other advantages of the disclosure will be apparent upon consideration of the following detailed description, taken in conjunction with the accompanying drawings, in which like reference characters refer to like parts throughout, and in which:

[0014] FIG. 1 is a block diagram depicting an online multi-browser system, according to an illustrative embodiment of the invention;

[0015] FIG. 2 is a block diagram depicting an exemplary computing device illustrated in FIG. 1;

[0016] FIG. 3 is a block diagram depicting an implementation of the system of FIG. 1, according to an illustrative embodiment of the invention;

[0017] FIG. 4 is a block diagram depicting an exemplary presentation application interface that may be presented across multiple browsers to a user;

[0018] FIG. 5 is a flow chart depicting the overall operation of the various components of the system of FIG. 1 implemented as an online presentation application;

[0019] FIG. 6 is a flow chart depicting the operation of the various components of the system of FIG. 1 when displaying speaker notes in a presentation application;

[0020] FIG. 7 is a flow chart depicting the operation of the various components of the system of FIG. 1 when the user navigates presentation slides using the parent browser window; and

[0021] FIG. 8 is a flow chart depicting the operation of the various components of the system of FIG. 1 when the user navigates presentation slides using the child browser window.

DETAILED DESCRIPTION OF ILLUSTRATED EMBODIMENTS

[0022] To provide an overall understanding of the invention, certain illustrative embodiments will now be described, including systems and methods for displaying a document on

a plurality of client computing devices by a server. However, it will be understood by one of ordinary skill in the art that the systems and methods described herein may be adapted and modified as is appropriate for the application being addressed and that the systems and methods described herein may be employed in other suitable applications, and that such other additions and modifications will not depart from the scope thereof.

[0023] Generally, the systems and methods described herein include software and hardware for controlling, from one browser window, the display of information in an online document processing application on another browser window. For example, the systems and methods described herein provide a user with the ability to view a presentation slide deck on a first browser window while simultaneously viewing speaker notes for each slide of the slide deck in a second browser window. The systems and methods not only provide the user with the ability to view speaker notes, but also control the slideshow from the second browser window. FIGS. 1 and 2 provide an overview of an exemplary system, computing devices, and methods used to provide such an online document processing application. FIGS. 3 and 4 describe in more detail one exemplary system having two browsers—a parent browser and a child browser—running an online presentation application. FIGS. 5-8 provide exemplary methods, according to the present disclosure, for controlling an online presentation application running in a parent browser window from either the parent browser window itself or from a child browser window.

GENERAL SYSTEM DESCRIPTION

[0024] FIG. 1 is a block diagram depicting an online multi-browser system 100, according to an illustrative embodiment of the invention. System 100 includes multiple clients 110 that can connect to servers, such as server 120, via a network 140. Network 140 may include a local area network (LAN), a wide area network (WAN), a telephone network, such as the Public Switched Telephone Network (PSTN), an intranet, the Internet, or a combination of networks. Three clients 110 and one server 120 are illustrated as connected to network 140 for simplicity. Two clients 110 are shown as being associated with a single user 111. In practice, there may be more clients, users and/or servers. Also, in some instances, a client may perform one or more functions of a server and a server may perform one or more functions of a client.

[0025] A client 110 may include a device, such as a personal computer, a laptop computer, a wireless telephone, a personal digital assistant (PDA), mobile devices, smart phones, touch-responsive tablet devices, or another type of computation or communication device. In certain embodiments, a client 110 may include display devices including a projection screen or television display. Users of clients 110 may access or receive information from server 120.

[0026] In the implementation shown, clients 110 may generally interact with server 120 such that clients 110, in conjunction with server 120, execute an online document application, such as an online presentation application. Server 120 may include software, labeled as document server component 125, to assist implementing the online presentation application. Online presentation applications created by users of clients 110 may be stored by server 120 in, for example, a database on storage media 130. Although illustrated as a single device in FIG. 1, server 120 may be implemented as, for example, a single computing device or as multiple distrib-

uted computing devices. One of ordinary skill in the art will appreciate that whether a device is functioning as a server or a client often depends on the specific application being implemented. That is, whether a computing device is operating as a client or a server may depend on the context of the role of the computing device within the application. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

[0027] The interaction of clients 110 with server 120 may be through browser programs 115 at clients 110. For example, the online presentation application may be a web application that runs within browsers 115. In this manner, clients 110 may not be required to install any presentation specific software to use the online presentation application at client 110. When browsers or browser programs are discussed herein, these terms are intended to refer to any program that allows a user to browse markup documents (e.g., web documents), regardless of whether the browser program is a stand alone program or an embedded program, such as a browser program included as part of an operating system.

[0028] FIG. 2 is an exemplary diagram of computing device 200, such as one of clients 110 or server 120. Computing device 200 may include a bus 210, a processor 220, a main memory 230, a read only memory (ROM) 240, a storage device 250, an input device 260, an output device 270, and a communication interface 280. Bus 210 may include a path that permits communication among the components of computing device 200.

[0029] Processor 220 may include any type of processor, microprocessor, or processing logic that interprets and executes instructions. Main memory 230 may include a random access memory (RAM) or another type of dynamic storage device that may store information and instructions for execution by processor 220. ROM 240 may include a ROM device or another type of static storage device that may store static information and instructions for use by processor 220. Storage device 250 may include a magnetic and/or optical recording medium and its corresponding drive.

[0030] Input device 260 may include a mechanism that permits a user to input information to computing device 200, such as a keyboard, a mouse, a pen, voice recognition and/or biometric mechanisms, etc. Output device 270 may include a conventional mechanism that outputs information to the user, including a display, a printer, a speaker, etc. Communication interface 280 may include any transceiver-like mechanism that enables computing device 200 to communicate with other devices and/or systems. For example, communication interface 280 may include mechanisms for communicating with another device or system via a network, such as network 140.

[0031] Document server component 125 may be implemented in software and stored in a computer-readable medium, such as memory 230. A computer-readable medium may be defined as one or more physical or logical memory devices and/or carrier waves.

[0032] The software instructions defining document server component 125 may be read into memory 230 from another computer-readable medium, such as data storage device 250, or from another device via communication interface 280. The software instructions contained in memory 230 cause processor 220 to perform processes that will be described later. Alternatively, hardwired circuitry may be used in place of or in combination with software instructions to implement pro-

cesses consistent with the present disclosure. Thus, implementations consistent with the principles of the systems and methods described are not limited to any specific combination of hardware circuitry and software.

### Online Presentation Application System

[0033] An online presentation application, as described herein, may be implemented as a distributed web application in which portions of the application execute at one or more of clients 110 and at server 120. More specifically, clients 110 that wish to use the online presentation application may request the presentation application from server 120. In response, server 120 may transmit portions of the presentation application for local execution at clients 110. The online presentation application may thus execute as a distributed application across server 120 and one or more of clients 110.

[0034] FIG. 3 is a block diagram depicting an implementation of the system 300 (or system 100 of FIG. 1), according to an illustrative embodiment of the invention. FIG. 3 depicts a client 110 having two browsers—a parent browser 350 and a child browser 352. In one example, when presenting, users may display the parent browser, having a slideshow, on a projector screen, while viewing a child browser from their laptop computer on a podium. As was described with reference to FIG. 1, system 300 may also include two or more clients, one running a parent browser and the others running child browsers. The parent browser 350 includes a client engine 316 and a user interface 317. The client engine 316 further includes a controller 318. The child browser 352 includes a user interface 319. The client 110 is connected via a network 140 to a document server component 125. Document server component 125 includes a front-end component 326 and a back-end component 327. Front-end component 326 may include a web and/or XML server that interfaces with clients 110. For example, front-end component 326 may receive HTTP requests from clients 110 and transmit XML data to clients 110. Back-end component 327 may perform the more substantive processing related to the presentation application, such as the calculation of functions required to update a set of slides.

[0035] In certain embodiments, portions of the online presentation application executes within clients 110. Client engine 316, within parent browser 350, may be implemented using, for example, Javascript that is downloaded from server 120 when needed by client 110. User interface 317 may provide the final user interface that is displayed in browsers 350 and 352 based on, for example, HTML (hyper-text markup language) and CSS (cascading style sheets) data supplied from client engine 316.

[0036] User interface 317 and client engine 316, together, act to reduce the start-stop start-stop nature of traditional browser-based web applications, as client engine 316 adds a client-side layer that can handle many of the user interactions with user interface 317. Instead of loading a web page at the start of a user session, browser 350 may load client engine 316 from server 120 (or, alternatively, from a local cache). Client engine 316 may be responsible for both rendering the interface the user sees and communicating with server 120 on the user's behalf. Client engine 316 allows the user's interaction with the online presentation application to happen asynchronously, i.e., independent of communication with server 120.

[0037] The client engine 316 further includes a controller 318, which is configured with code to create the child browser 352 and exchange data, including but not limited to HTML,

CSS, Javascript, with the user interface 319 of child browser 352. In certain embodiments, the controller 318 may receive HTML and CSS data to be displayed in the child browser 352 from the server 120 (or, alternatively, from a local cache).

[0038] During operation, the client engine 316 initiates a user session of the presentation application within the parent browser 350. Upon a request from a user, via user interface 317, the controller 318, may open a child browser 352 for displaying speaker notes and other information to the user via user interface 319. In certain implementations, the controller 318 may communicate directly or indirectly (via network 140) to the child browser 352. The user may, via user interface 319 in child browser 352, choose to control the progression of slides displayed in the parent browser. The controller 318 is programmed to subscribe to events such as button clicks or keyboard presses while the user is on the child browser 352. Upon detection of such an event, the controller updates the slides in the parent browser 350, and the child browser 352 updates the speaker notes.

[0039] In one implementation, the client-side operation of the online presentation application may be implemented using the known Asynchronous JavaScript And XML (AJAX) web development techniques, although one of ordinary skill in the art will recognize that other technologies could be used.

[0040] Although, the parent browser 350 and child browser 352 are shown, in FIG. 3, as being within a single client device 110, in certain implementations, the parent browser 350 and the child browser 352 may be on different clients. In such implementations, referring to system 100 shown in FIG. 1, parent browser 350 may include browser 115 in client 110 and child browser 352 may include browser 115 located in client 110 operated by user 111. In such implementations, the parent and child browsers are in separate client devices. In one example, a user may use their mobile device, smart phone, tablet or any suitable device for controlling a presentation being displayed on a different computing device. The different computing device may include another computer or laptop or any suitable computing device that may or may not be connected to a projection screen for displaying the presentation. In certain implementations, the different computing device is embedded within a projection screen. The different computing device may also include another mobile device, smart phone, tablet or any other suitable device so that while one device is viewing the presentation, another device is controlling the progression of the presentation slides.

[0041] In certain implementations, during operation of a system having parent and child browsers on different devices, the client engine 316, instead of communicating directly with the child browser 352, communicates via the network 140 and server 120. The client engine 316 may initiate a user session of the presentation application within the parent browser 350. Upon a request from a user in local or remote client device, via user interfaces 317 or 319, the server 120, may open a child browser 352 for displaying speaker notes and other information to the user via user interface 319. The user may, via user interface 319 in child browser 352 on a different client device, choose to control the progression of slides displayed in the parent browser 350.

[0042] The parent browser 350 may, itself, be initiated by the server 120. In such implementations, upon receiving a request from one or more client devices, the server 120 initiates a parent browser 350 on a client device to run the presentation application. Upon receiving a request from an

authorized user, the server 120 may initiate a child browser 352 on the same client device or a different client device. The server 120 may request additional authorization to allow a user to operate the child browser 352 and thereby control the presentation being displayed on the parent browser 350.

[0043] The authorized user may include at least one of a user operating a client device having the parent browser 350, a user operating a client device intended to have the child browser 352, a user operating a client device having the child browser 352, and a user authorized by a user of the client device having the parent browser 350. The authorized user may generally include any user possessing suitable credentials to view or run the presentation.

[0044] In certain implementations, a user may initiate the child browser 352 from a location remote to the location of the parent browser 350. For example, a first user may be viewing a presentation remotely via videoconference. The presentation may be running on a different computing device operated by a second user. In such an example, the second user may send a link to the first user to allow the first user to access and open a child browser 352 in the first user's remote location. The first user may then, via the child browser 352 control the slides in the presentation.

[0045] Additionally and optionally, the systems and methods described herein may be adapted to allow for a plurality of parent browsers 350 and a plurality of child browsers 352. In such arrangements, one child browser 352 may be configured to operate presentations running on a plurality of parent browsers 350. The child browser 352 may control the operation of the same or different presentations running on the plurality of parent browsers 350. For example, the child browser 350 may function as a dashboard having user interface elements for simultaneously control a plurality of different presentation slide decks. In another arrangement, a plurality of child browsers 352 may be configured to control a single parent browser 350. In such an arrangement, a plurality of users collaborating on a presentation slide deck may have child browsers 352 capable of controlling the presentation running on a single browser 350. Generally, one or more child browsers 352 in one or more client devices may be configured to control one or more parent browsers 350 in one or more of the same or different client devices.

#### User Interface

[0046] FIG. 4 is a block diagram depicting an exemplary presentation application interface 400 that may be presented across multiple browsers to a user. Interface 400 depicts the interface displayed to the user when presenting a slide show. In particular, interface 400 includes a parent browser window 450 and a child browser window 452. The parent browser 450 includes slides 402, a progress button 404, slide selector menu 406 and an action menu 408. The action menu 408 includes a list of user-selectable items including "Show speaker notes" item 412. The parent browser 450 may include any number of text boxes, media placeholders, forms, buttons, drop-down menus, radio buttons, check-boxes as desirable without departing from the scope of the invention. When displayed, users may view and modify one or more slides in their presentation.

[0047] The child browser window 452 depicts speaker notes 430 along with other useful information for the user. In particular, the child browser window 452 includes slide preview section 415 having a slide preview 420, a previous slide preview/selector 422, a next slide preview/selector 424 and a

slide selector menu **426**. The child browser window **452** further includes a timing section **428** having a time indicator, a pause button and a reset button. The user interface may include any number of browsers configured to control each other. The user interface may also include any number of input components (e.g., buttons) and output components (e.g., slides) without departing from the scope of the invention.

#### Methods of Operation

**[0048]** Turning to FIGS. **5-8**, the methods of operation of system **100** of FIG. **1** (and system **300** of FIG. **3**) will be described in more detail. FIG. **5** is a flow chart depicting the overall operation of the various components of the system of FIGS. **1** and **3** implemented as an online presentation application. A user at one of clients **110** may initiate a presentation application session (step **501**). For instance, the user of a client **110** may contact front-end component **326** using parent browser **352**. In some implementations, the user may be required to have previously registered or created an account with and be logged into document server component **125**.

**[0049]** In response to the user initiating a presentation session, the document server component **125** may transmit client engine **316** to the client **110** (step **502**). Some portions of client engine **316** may be cached at client **110** from a previous session, in which case these portions do not need to be re-downloaded from server **120**. In some implementations, the programming code that comprises client engine **316** may be implemented in a modular manner. In this situation, portions of client engine **316** may be transmitted to client **110** on an as-demand basis as the functionality of those portions of client engine **316** are needed by client **110**. For example, client engine **316** may implement a number of “core” functions, such as basic user interaction and display functions, that are always transmitted to client **110**, while other functions, such as more specialized presentation functions, may be transmitted to client **110** on an as-need basis. Additionally, some functions of online presentation application may always be executed by back-end component **327** and may therefore never be transmitted to client engine **316**.

**[0050]** Because client engine **316** can be transmitted to client **110** as needed by client **110**, the online presentation application may be an entirely web-based application in which the client does not need to pre-install any portion of client engine **316**. Advantageously, a user will typically be able to use the online presentation application from any computing device that includes a compatible browser **115** and that is connected to network **140**.

**[0051]** Client engine **316**, once loaded at client **110**, may present, via user interface **317**, the graphical user interface (interface **400** of FIG. **4**) of a slide deck to the user (step **503**). The slides **450** are the data object that the presentation application presents to the user. The slide deck presented to a user may be a “blank” slide deck from which the user may begin a new presentation document, a presentation document retrieved from a previous session of the online presentation application, or a version of the presentation document imported from a conventional presentation document (e.g., an “.pptx” document, a “.key” document, or other types of known presentation document formats).

**[0052]** Through interface **400** of FIG. **4**, a user can, for example, enter text and media into the presentation slides and perform formatting operations relating to the slide deck. In response, client engine **316** interacts with and makes requests

of document server component **120** as necessary (step **504**). The above-described functions and operations that are allowable by the online presentation application are exemplary. Through the interface **400**, the online presentation application may permit a user to perform numerous operations that are generally recognized as being consistent with a presentation application.

**[0053]** In one implementation, client engine **316** may handle most or all of the functionality associated with responding to user actions relating to navigating and/or formatting the interface without needing to contact document server component **125**. Other functions of the presentation application may be handled by logic in back-end component **327**. By handling formatting and data entry feedback locally at client **110**, the online presentation application can provide a high level of responsiveness to user actions that tend to occur frequently and for which users expect immediate feedback.

**[0054]** One of ordinary skill in the art will recognize that in alternate implementations, the division of functions executed at client engine **316** and back-end component **327** may be modified such that client engine **316** handles more or fewer functions and similarly, back-end component **327** may also handle more or fewer functions.

**[0055]** Periodically, or at the explicit direction of the user via interface **400**, the data that comprises the presentation document may be saved (step **505**). In one possible implementation, the presentation document may be saved by document server component **125** to database **130**. In this manner, the presentation document may be accessible by the user (or by other users) from any computing device connected to network **140**.

**[0056]** As described above, the user may interact and select from several options in interface **400** in FIG. **4**. For example, at the start of a presentation, the user may choose to view speaker notes in a separate browser window. Accordingly, the user may select the “Show speaker notes” button **412** in the action menu **408** in the parent browser window **450**.

**[0057]** FIG. **6** shows the various steps that the system performs when a user selects the show speaker notes option (step **601**). In particular, FIG. **6** is a flow chart depicting the operation of the various components of the system of FIG. **1** when displaying speaker notes in a presentation application. The controller **318** in the client engine **316** runs code to open a “blank” new child browser window **452** (step **602**). If a window was already opened, the controller **318** may close the open window and then open a new child window. In certain embodiments, the controller **318** may select a unique name when opening a new child window.

**[0058]** Once the new child window is opened, the controller **318** populates the new child window with HTML and CSS from the server (step **603**) or a local cache. In particular, the HTML and CSS data includes, among other things, the contents of the speaker notes to be displayed in section **430**, the contents of the slides to be displayed in section **415** and other information related to the arrangement and appearance of the child window **452**.

**[0059]** Additionally, the controller **318** provides the child browser window **352** information about event targets (step **603**). Events may include operations involving user interactions with the user interface in the parent browser or child browser. Typical events may include inputs from a mouse or pointing device, keyboard or typing device, HTML frame/object, HTML form, and other user interfaces. Some

examples of inputs from a mouse include double-clicks, mouse down, mouse up, mouse over, mouse move and mouse out. Some example of inputs from a keyboard include key down, key press, key up.

[0060] In certain embodiments, an event target may relate to an event such as selecting with a mouse, clicking-on a button, or using a keyboard shortcut in the parent browser window 450 or child browser window 452. Event targets in the parent browser window 450 may include slide progress button 404, or an item from the slide selector menu 406. In certain embodiments, mouse clicks anywhere on the slide 402 may be an event target to indicate slide advancement. Similarly, event targets in the child browser window 452 may include buttons 422 or 424 for moving slides forward or backward, slide selector 426, or any other portion of the child window 452. Generally, event targets may be assigned to any portion of the parent and child browser windows.

[0061] Returning to FIG. 6, the child window 352 may include code for listening to parent window 350 event targets and child window event targets. The child window 352 may begin listening for parent window event targets (step 604) using an event handler script. The parent browser, particularly the controller 318, also subscribes to child window event targets (step 605).

[0062] During use of the online presentation application, a user may select the parent browser window or the child browser as being active. FIG. 7 depicts the various steps that the system performs when the user selects the parent browser window as being active (step 701). In particular, the user may click on an event target. For example, the user may desire to advance the slide and click on the slide progress button 404 (step 702). The controller 318, listening for such an event, may read this button click as an event and run the corresponding Javascript to advance the slides in the parent browser window 450 (step 703). The child browser, also listening for such an event in the parent window (as described with reference step 604 in FIG. 6), initiates changes within the child browser window to advance the slide in the preview section 415 and update the speaker notes section 430 to correspond to the new slide (step 704). The child window may also update its other components to conform to the new slide and any passage of time.

[0063] In certain embodiments, during use of the online presentation application, a user may select the child browser window as being active. FIG. 8 depicts the various steps that the system performs when the user desires to control the parent browser window from the child browser window. An example of such situation is when a user is reviewing speaker notes while presenting the slide deck. The speaker may be standing at a podium delivering a presentation to an audience. The audience may be able to view the slide deck (parent browser window 450) on a large projection screen while the user is viewing the speaker notes (child browser window) on a laptop at the podium. In such an example, the slides are typically running from the same laptop and the user may be controlling the advancement of the slides. During the presentation, the user may desire to view the speaker notes and select the child window, thereby automatically deselecting the parent browser window (step 801).

[0064] In certain embodiments, when the user is ready to advance a slide, the user may select an event target in the child window (step 802). For example, the user may select button 424 in child window 452 to move from slide 2 to slide 3. When the user selects such an event target, the child browser causes

an action to be initiated which the controller 318 is subscribed to listen for (step 803). Consequently, the controller 318 updates runs one or more Javascript functions and updates the parent browser window to advance the slides (step 804). The child window also updates its components to advance the slide in the preview section 415 and update the speaker notes section 430 to correspond to the new slide (step 805). Advantageously, according to systems and methods described herein, the child browser window does not need to include Javascript for controlling slides. Instead slide progression and control is handled by the controller 318 which is subscribed to listen to child window event targets such as keyboard shortcuts and mouse clicks.

[0065] It will be apparent to one of ordinary skill in the art that aspects of the present disclosure, as described above, may be implemented in many different forms of software, firmware, and hardware in the implementations illustrated in the figures. The actual software code or specialized control hardware used to implement aspects consistent with the principles of the disclosure is not limiting of the invention. Thus, the operation and behavior of the aspects of the disclosure were described without reference to the specific software code—it being understood that one of ordinary skill in the art would be able to design software and control hardware to implement the aspects based on the description herein.

[0066] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous.

[0067] Further, certain portions of the disclosure may be implemented as “logic” or a “component” that performs one or more functions. This logic may include hardware, such as an application specific integrated circuit or a field programmable gate array, software, or a combination of hardware and software.

1. A computerized method for rendering and controlling a presentation document in an online presentation application, comprising:

- transmitting, from a client computing device to a server, a request to initiate a presentation application session;
- in response to the request, receiving a client engine from the server;
- displaying in the session, using the client engine and on a first web browser window at the client computing device, a first portion of an online collaboratively editable presentation document;
- receiving, at the client engine and via a second web browser window at the client computing device, a request for displaying a second portion of the online collaboratively editable presentation document on the first browser; and
- replacing, using the client engine and in the first web browser window of the client computing device, the displayed first portion of the presentation document with the second portion of the presentation document based on the request received at the second browser window; wherein the client engine is configured to listen for requests received in the second web browser window.

2. The computerized method of claim 1, wherein each of the first portion and second portion of the presentation document includes one or more presentation slides.

3. The computerized method of claim 1, further comprising displaying, using the client engine and on the second browser window at the client computing device, one or more user input components for allowing users to submit a request.

4. The computerized method of claim 3, wherein the one or more user input components include at least one of an image, button, text box, radio-button, and check-box.

5. The computerized method of claim 1, further comprising displaying, using the client engine and on the second browser window at the client computing device, a third portion of the presentation document.

6. The computerized method of claim 5, wherein the third portion of the presentation document includes one or more pages of notes.

7. The computerized method of claim 5, further comprising receiving, at the client engine and via the first browser window at the client computing device, a request for displaying the third portion of the presentation document in the second browser window.

8. The computerized method of claim 1, wherein the first portion of the presentation document is replaced with the second portion of the presentation document in the first browser window based on the request received at the client engine.

9. The computerized method of claim 1, further comprising opening, by the client engine, the second browser window in response to a request received at the first browser window.

10. The computerized method of claim 1, further comprising displaying, using the client engine and via the first browser window, a third portion of the presentation document including one or more pages of notes.

11. A computerized method for rendering and controlling Hypertext Markup Language (HTML) data and Cascading Style Sheets (CSS) data in a browser, comprising:

transmitting, from a client computing device to a server, a request to initiate a session;

in response to the request, receiving a client engine from the server;

receiving, at the client engine and via a first browser window at the client computing device, a first set of data including HTML, CSS and Javascript data;

receiving, at the client engine and via the second browser window at the client computing device, a request for replacing the first set of data with a second set of data including HTML, CSS and Javascript data; and

replacing, by the client engine in the first browser window of the client computing device, the first set of data with the second set of data based on the request received in the second browser window;

wherein the client engine is configured to listen for requests received in the second browser window.

12. The computerized method of claim 11, wherein the client engine receives the second set of data from a server and creates the second browser window.

13. The computerized method of claim 11, wherein the first set of data is replaced with the second set of data based on a request in the first browser window.

14. The computerized method of claim 11, further comprising displaying, using a client engine and on the second browser window at the client computing device, one or more user input components for allowing users to submit a request.

15. The computerized method of claim 11, further comprising receiving, at the client engine and via the second browser window at the client computing device, a third set of data including HTML and CSS data.

16. A system having circuitry for rendering and controlling a presentation document in an online presentation application at a client computing device, comprising:

- a client computing device including a first browser window having a client engine for running the online presentation application, and a user interface for displaying a portion of the presentation document to a user and receiving requests from the user, wherein the client engine is received from a server in response to a request to initiate a presentation application session; and
- a client computing device including a second browser window having a user interface for receiving requests from a user for controlling the presentation document displayed in the first browser window;

wherein the client engine includes a controller circuit for generating the second browser window, listening to requests received at the second browser window, and replacing the displayed portion of the presentation document in the first browser with another portion of the presentation document based on the received requests.

17. The system of claim 16, wherein the client engine is configured to provide HTML, CSS and Javascript data to the user interface in the first browser.

18. The system of claim 16, wherein the client engine is configured to receive Javascript calls from the user interface in the first browser.

19. The system of claim 16, wherein the client engine is configured to receive data corresponding to the presentation document and online presentation application data from a server.

20. The system of claim 16, wherein the user interface in at least one of the first browser and the second browser includes components receptive to inputs received from at least one of a mouse, keyboard, HTML frame/object, and HTML form.

21. The system of claim 20, wherein the components are receptive to at least one of mouse double-clicks, mouse down, mouse up, mouse over, mouse move, mouse out, key down, key press, and key up.

\* \* \* \* \*