March 30, 2015

Providing Event Notifications

Ravi Mandala

Providing Event Notifications

Abstract

An event notification system can be used to provide notifications to a user related to an event. The system can provide these notifications before, during, and after the event. The event notification system analyzes an upcoming event from a user’s calendar to determine a context information associated with the event. As an example, this context information can include location of the place where the event is scheduled, people attending the event, duration of the event, agenda associated with the event etc. Based on the determined context associated with the event, the system determines specific tasks associated with the event. For example, these tasks can include providing the user with a map with the coordinates of the location of the event, names of people who have not yet RSVP'd to the event etc. Subsequently, the system provides notifications to the user for the specific tasks associated with the event.

Problem Statement

Current calendar applications provide basic notifications to users about events scheduled on the users’ calendars. The notifications simply notify the user of the event before the event is to occur. For example, a calendar application can provide a notification to the user about an upcoming event at a predefined time prior to the occurrence of the event. The calendar application might also provide a second notification at a second predefined time after the first notification. However, these calendar applications do not reminder the user of various tasks that may be associated with a scheduled calendar event. In order to make the notifications for an event more useful and helpful to the user, there is a need to provide notifications to the user for tasks associated with a scheduled calendar event based on the context of the event. These notifications can be provided to the user before, during, or after the event in order to make the notifications helpful and meaningful.

Event Notification System

The systems and techniques described in this disclosure relate to an event notification system. The system can be implemented for use in an Internet, an intranet, or another client and server environment. The system can be implemented locally on a client device or implemented across a client device and server environment. The client device can be any electronic device such as a mobile device, a smartphone, a tablet, a handheld electronic device, a wearable device, a laptop, etc.

Fig. 1 illustrates an example method 100 of providing notifications to a user for an event scheduled on the user's calendar application. Method 100 can be performed by the event notification system.

A user can have multiple events scheduled on his/her calendar. The user can schedule events himself on his calendar, e.g., the user knows he has to pick up his dry cleaning on Tuesday at 9 a.m. and schedules a calendar event for the same time. Alternatively, or additionally, others can schedule events on the user's calendar, e.g., the user's friend invites the user to a BBQ on Tuesday at 6 p.m. and schedules the event on the user's calendar for the same time. The events on the user's calendar can be distributed along a period of time. The events could range from being scheduled for the next hour, to being scheduled for a day or a year in the future. As an example, these events can include an office meeting scheduled for next day, or a football game the user has scheduled with his friends over the weekend. Scheduled events could also include all-day events, e.g., a full-day seminar on “Defensive Publications” scheduled for next Monday that the user has to give at a college.

The event notification system analyzes an upcoming event from the user’s calendar to determine a context information associated with the event (block 110). The context information can be defined as any kind of information pertaining to the event. The context information can vary depending on the detected event. For example, the context information can include the location of event, the participants attending the event, the time of the event, the type of the event, whether the user is a presenter for the event, an agenda for the event, actions items for the event, requirements for the event, etc. There are various ways the system can determine the context information for a calendar event. The system can determine the context information from the information in the calendar event itself. This information can be input when the calendar event is created. For example, when a calendar event is created, the creator provides a location, time, and participants for the event. Other information can also be provided such as an agenda or action items for the event. The system can analyze the calendar event information to generate the context information for the event. Alternatively, or additionally, the system can prompt the user to identify specific contexts for the calendar event. For example, the system can provide the user with a list of contexts. The contexts may be pre-curated by the system based on one or more ranking algorithms. The user can then help the system identify contexts for the event. One way of determining context information by the system can be using an optical character recognition algorithm in combination with a server which includes history of the user interactions, user's previous responses to different event requests, user's calendar information etc. The system can use the optical character recognition to read information from the events and use this information along with the information in the server to determine the context information. Another way of determining the context information can be using machine learning tool algorithms, which keep on learning and improving, based on the user's responses, preferences, inputs etc.

The system can be automated to analyze different events scheduled in the user's calendar at a certain time of the day or can be automated to check for any new events on a regular basis such as hourly, weekly, monthly, etc. Additionally, or alternatively, the system can analyze a particular event in response to a manual input by the user, e.g., when the user causes the system to analyze calendared events for context information. Additionally, or alternatively, as soon as a new event is added to the user's calendar application, the system analyzes the event to determine the context information associated with the event. After the context information associated with an event is determined, the system determines specific tasks associated with the event based on the context of the event (block 120). The system can determine the specific tasks for a determined context based on a predetermined mapping of contexts with specific tasks. A human-curated list can be provided to the system that associated specific tasks with specific contexts. Alternatively, or additionally, a list of specific tasks can be generated for contexts using machine learning techniques. Alternatively, or additionally, a list of specific tasks can be generated for contexts using a combination of human-curating and machine learning techniques. For example, the system can map a calendar event with a "meeting" context to the specific tasks of "Create an agenda," "Begin wrapping up the meeting," "Send out meeting minutes," and

"Set-up follow-up meetings." As a further example, the system can map a calendar event with a "presenting" context to the specific tasks of "Prepare slides," "Practice presentation," "Begin presenting slides," and "Distribute slides." The system can determine multiple contexts for a calendar event and combine specific tasks for the multiple contexts for the calendar event. For example, if the user has an office meeting scheduled for next day, the system can determine associated tasks based on the context information associated with the office meeting. In this example, the system determines the context to be that the user is going to the meeting, the user is an attendee, the user doesn't have to present anything, the agenda of the meeting, the name of the conference room where the meeting is going to be held, the action items, and to bring a printout of the slides before coming. Based on this determined context information, the system determines specific tasks such as reminding the user to bring the printouts of the slides and providing the location of the conference room.

Additionally, the system schedules reminders for these specific tasks associated with the event at a particular time e.g., at a time before the event. The system can determine the particular time to remind of these specific tasks, e.g., from particular times pre-determined for each specific task or based on an algorithm which scans the analyzed event for any timeline present within the event, or scan's user's email client to determine the required information associated with the event, etc. The reminders can be provided to the user before the event, during the event, or after the event. In an example, the system determines the specific task of sending a request for feedback for a meeting. The system schedules a notification reminder to send the request for feedback one day after the In yet another example, the system determines the specific task of providing a reminder to the presenter to wrap up the meeting. The system schedules a notification reminder to notify the present to wrap up the meeting 10 minutes before the end of the scheduled meeting time.

Additionally, or alternatively, the system can also be capable of setting up template reminders such as a "start now" reminder which suggests to head towards the meeting location, a "prepare slides" reminder for whoever is going to present something, a "invite more relevant folks" reminder for any organizer of the event, or a "start wrapping up now" reminder to remind the user that the end of the event is approaching.

Subsequently, the system provides notifications to the user for the specific tasks (block 130). These notifications can be provided to the user at the time scheduled by the system. The notifications can be provided to a client device associated with the user. The notification can be a message notification describing the nature of the reminder, e.g., "prepare slides" or "wrap-up meeting," displayed on a display screen associated with the client device. Alternatively, or additionally, the client device can emit sounds associated with the notification, e.g., a verbal readout describing the nature of the reminder, from a speaker associated with the client device. Alternatively, or additionally, the notification can be vibrations emitted by the client device.

FIG. 2 is a block diagram of an exemplary environment that shows components of a system for implementing the techniques described in this disclosure. The environment includes client devices 210, servers 230, and network 240. Network 240 connects client devices 210 to servers 230. Client device 210 is an electronic device. Client device 210 may be capable of requesting and receiving data/communications over network 240. Example client devices 210 are personal computers (e.g., laptops), mobile communication devices, (e.g. smartphones, tablet computing devices), set-top boxes, game-consoles, embedded systems, and other devices 210' that can send and receive data/communications over network 240. Client device 210 may execute an application, such as a web browser 212 or 214 or a native application 216. Web applications 213 and 215 may be displayed via a web browser 212 or 214. Server 230 may be a web server capable of sending, receiving and storing web pages 232. Web page(s) 232 may be stored on or accessible via server 230. Web page(s) 232 may be associated with web application 213 or 215 and accessed using a web browser, e.g., 212. When accessed, webpage(s) 232 may be transmitted and displayed on a client device, e.g., 210 or 210'. Resources 218 and 218' are resources available to the client device 210 and/or applications thereon, or server(s) 230 and/or web pages(s) accessible therefrom, respectively. Resources 218' may be, for example, memory or storage resources; a text, image, video, audio, JavaScript, CSS, or other file or object; or other relevant resources. Network 240 may be any network or combination of networks that can carry data communication.

The subject matter described in this disclosure can be implemented in software and/or hardware (for example, computers, circuits, or processors). The subject matter can be implemented on a single device or across multiple devices (for example, a client device and a server device). Devices implementing the subject matter can be connected through a wired and/or wireless network. Such devices can receive inputs from a user (for example, from a mouse, keyboard, or touchscreen) and produce an output to a user (for example, through a display). Specific examples disclosed are provided for illustrative purposes and do not limit the scope of the disclosure.

DRAWINGS

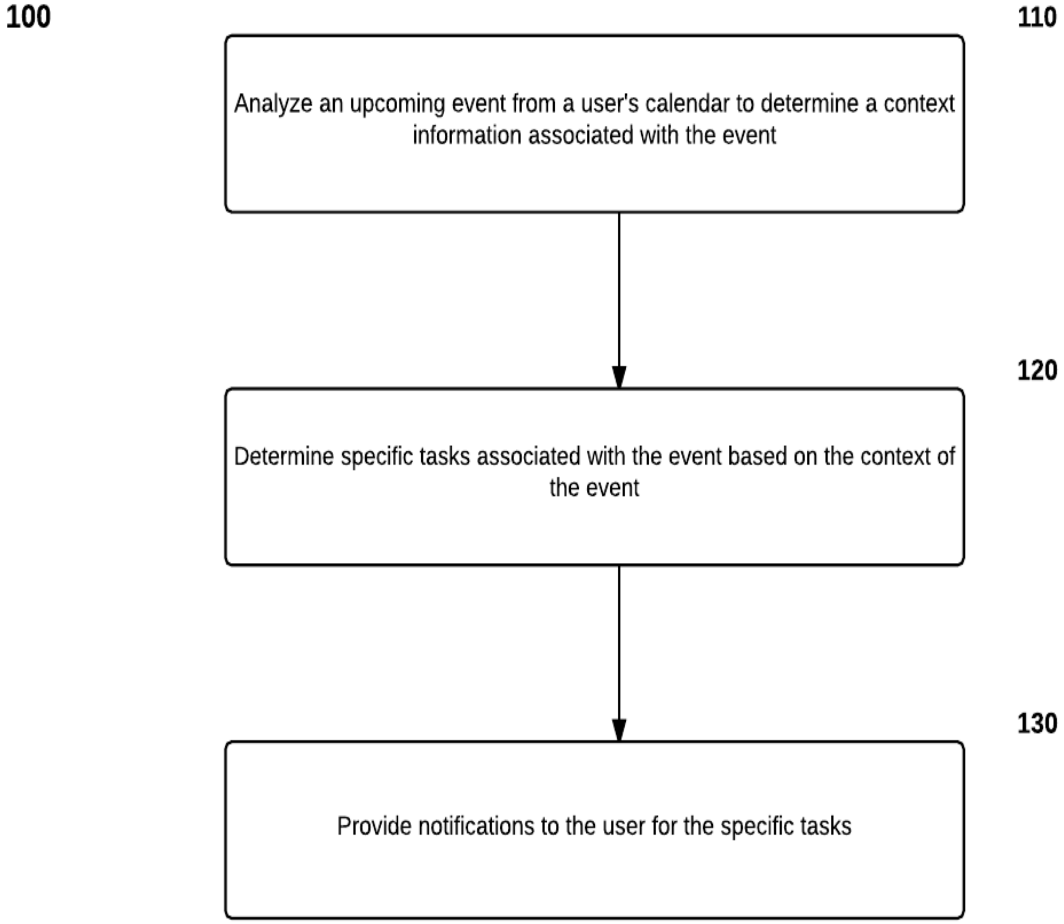
100
110
Analyze an upcoming event from a user's calendar to determine a context information associated with the event
120
Determine specific tasks associated with the event based on the context of the event
130
Provide notifications to the user for the specific tasks

FIG. 1

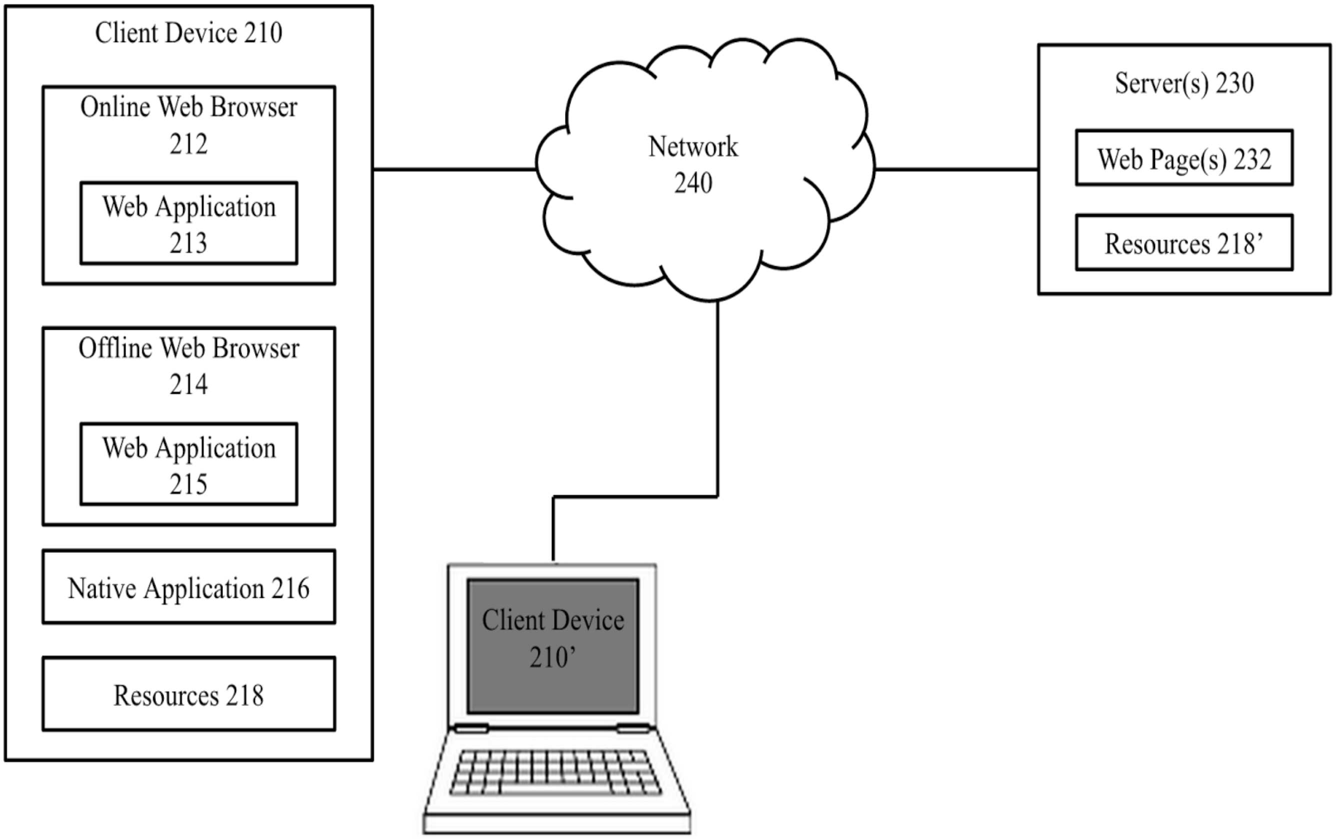
Client Device 210
Online Web Browser 212
Web Application 213
Offline Web Browser 214
Web Application 215
Native Application 216
Resources 218
Network 240
Server(s) 230
Web Page(s) 232
Resources 218'
Client Device 210'

Fig. 2