(51) International Patent Classification:
G06Q 10/10 (2012.01)

(21) International Application Number:
PCT/US20 16/049409

(22) International Filing Date:
30 August 2016 (30.08.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/842,097 1 September 2015 (01.09.2015) US


(54) Title: CONSENSUS SCHEDULING FOR BUSINESS CALENDAR

(57) Abstract: Using a set of alternative events with both modern and legacy systems. A method includes identifying a master message including a set of alternative events formatted for a modern system. The master message includes default information for all events in the set of alternative events as well as exception information representing exceptions to the default information for individual events in the set of alternative events. A plurality of instance messages is identified. Each of the instance messages includes default information from the master message and exception information for a given event in the set of alternative events. The master message and the instance messages are sent to other systems without regard to whether the other systems are modern or legacy systems so as to allow modern systems to select event alternatives using the master message and to allow legacy systems to select event alternatives using the instance messages.
Declarations under Rule 4.17:

— as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(H))

— as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(H)))

Published:

— with international search report (Art. 21(3))
CONSENSUS SCHEDULING FOR BUSINESS CALENDAR

BACKGROUND

Background and Relevant Art

[0001] Computers and computing systems have affected nearly every aspect of modern living. Computers are generally involved in work, recreation, healthcare, transportation, entertainment, household management, etc.

[0002] As computer technology advances, new features may be added to new (referred to herein as modern) versions of existing systems. As these features are added, there may be older (referred to herein as legacy) versions of the existing systems that are not able to natively implement the new features. However users of these legacy versions of systems may wish to take advantage of the new features in the modern versions of the systems.

[0003] For example, modern versions of scheduling systems (such as the calendar functionality included in Microsoft Exchange Server and Microsoft Outlook client available from Microsoft Corporation of Redmond, Washington) may include functionality that allows advanced scheduling features, such as the ability to have exceptions for appointments in a series of appointments, modify individual appointments in a series of appointments, add additional appointment instances to a series of appointments, collaborate on appointment details, etc. In some situations a server may have this functionality enabled and modern clients can make use of the functionality while legacy clients are unable to make use of the functionality, even though the server supports it. While some legacy systems allow for some of this functionality as well, later changes to a series will destroy any exceptions. This can create difficulties for users of both the modern clients and the legacy clients. In particular, a user at a modern client may utilize some of the functionality of the modern server and expect other users, including users at legacy clients, to be aware of the utilization. For example, a user at a modern client may update an instance of a series of appointments. Other users using modern clients would be made aware of the update, but users on legacy clients may not be made aware of the update, or may be made aware of the update in a way that breaks the series of appointments as a series. It would be useful to implement systems where modern and legacy clients could both implement new functionality and still be able to interact with one another.

[0004] The subject matter claimed herein is not limited to embodiments that solve any disadvantages or that operate only in environments such as those described above. Rather,
this background is only provided to illustrate one exemplary technology area where some embodiments described herein may be practiced.

BRIEF SUMMARY

[0005] One embodiment illustrated herein includes a method that includes acts for facilitating using a set of alternative events with both modern and legacy systems. The method includes identifying a master message including a set of alternative events formatted for a modern system. The master message includes default information for all events in the set of alternative events as well as exception information representing exceptions to the default information for individual events in the set of alternative of events. A plurality of instance messages is identified. Each of the instance messages includes default information from the master message and exception information for a given event in the set of alternative events. The master message and the instance messages are sent to other systems without regard to whether the other systems are modern or legacy systems so as to allow modern systems to select event alternatives using the master message and to allow legacy systems to select event alternatives using the instance messages.

[0006] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0007] Additional features and advantages will be set forth in the description which follows, and in part will be obvious from the description, or may be learned by the practice of the teachings herein. Features and advantages of the invention may be realized and obtained by means of the instruments and combinations particularly pointed out in the appended claims. Features of the present invention will become more fully apparent from the following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] In order to describe the manner in which the above-recited and other advantages and features can be obtained, a more particular description of the subject matter briefly described above will be rendered by reference to specific embodiments which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments and are not therefore to be considered to be limiting in scope, embodiments will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:
Figure 1 illustrates a server configured to perform consensus scheduling;
Figure 2A illustrates a server and system for propagating values from a master message to instance messages;
Figure 2B illustrates instance messages being updated;
Figure 2C illustrates instance messages being updated;
Figure 2D illustrates instance messages being updated;
Figure 3 illustrates instance messages being updated;
Figure 4A illustrates instance messages being updated;
Figure 4B illustrates instance messages being updated; and
Figure 5 illustrates a method for facilitating using a set of alternative events with both modern and legacy systems.

DETAILED DESCRIPTION

Some modern calendaring systems may allow for consensus scheduling for calendar events. In particular, with reference to Figure 1, modern calendaring system servers, such as the server 102 may be able to generate a meeting invite, shown as master message 104 with various alternatives (E1, E2, and E3) (while three alternatives are shown here, it should be appreciated that any number of alternatives may be used) for the event. For example, the meeting invite may ask recipients to choose between different alternatives in terms of different times for a given event, different locations for the given event, different subjects for the given event, different required attendees for the given event, etc. The master message 104 can be sent to users on modern clients 108-A. At the modern clients 108-A, the master message 104 can be used by each of the clients to render the alternatives on a user interface at the modern clients 108-A. For example, a user at the modern client 108-1 may see an email message rendered at the client 108-1 in a graphical user interface with user selectable links that allow the user at the client 108-1 to select one of the alternatives. The user can select a link to select an alternative, which causes the response 114 to be returned to the server 102.

The users on the modern clients 108-A can respond to the master message with an appropriate response 114 to choose a preferred alternative (or rank alternatives) from among the different alternatives. In particular, because both the modern server 102 and the modern clients 108-A understand the semantics of the master message 104 and the response 114, each client only needs to receive the master message and respond with a single response 114. Note that while Figure 1 illustrates only a single instance of the master message 104 and response 114, it should be appreciated that the master message will be sent to each of
the clients 108-1, 108-2, 108-3, 108-4, and any additional clients included in the set of modern clients 108-A, and that each client in the set of modern clients 108-A will send their own individual response 114 with different preference information.

[0020] Each user's choice can be returned to the calendaring server 102 which can then either automatically choose a consensus alternative or can allow a meeting organizer to select a consensus alternative based on all of the individual responses to the master message 104.

[0021] However, legacy clients 108-B may not include the ability to interpret the alternative information in the master message 104, and therefore, such an invite will not be usable by the legacy clients 108-B to indicate a preference for an alternative. Embodiments described herein allow multiple alternative legacy meeting invites, illustrated as instance messages 106, to be created from the master message 104. These instance messages 106 are simply legacy invite messages where each one includes one of the alternatives. As illustrated, message 106-1 includes the alternative A1, message 106-2 includes the alternative A2, and message 106-3 includes the alternative A3 in place of the default value D4. Each of the instance messages 106 is sent to each of the legacy clients as a legacy meeting invite. Thus, for example, client 108-5 will receive the messages 106-1, 106-2, and 106-3 as individual meeting invites. Similarly, client 108-6 will receive the messages 106-1, 106-2, and 106-3 as individual meeting invites. Users at the legacy clients 108-B can then respond to each invite individually, either accepting or declining each invite, as illustrated by the responses 118.

[0022] The server 102 can then use responses 118 from the legacy clients 108-B for identifying a consensus meeting. Notably, the server 102 can use the responses 118 from the legacy clients 108-B along with the responses 114 from the modern clients 108-A to identify a consensus meeting for all clients, both modern clients 108-A and legacy clients 108-B.

[0023] To alert the modern clients 108-A to the selected consensus meeting, the server 102 can simply send a confirmation message 120 to each of the modern clients 108-A which will cause the consensus meeting to appear on calendar user interfaces as the modern clients 108-A and/or to remove any of the non-selected instances from the user's calendar interface.

[0024] For the legacy clients 108-B, a number of various alternatives may be implemented for messages 122 sent to the legacy clients 108-B. For example, in some embodiments, the server 102 could send cancellations in the messages 122 for any meetings not selected by the server 102 as the consensus meeting alternative. In some embodiments,
this could be selectively performed. In particular, the server 102 could determine if any
decline messages had been sent for meeting alternatives that were not eventually selected as
the consensus meeting. In the case where declines were already sent, there would be no need
to send a cancellation message.

[0025] In some embodiments, the server 102 may need to resend invites in the messages
122 to one or more of the legacy clients 108-B for the meeting alternative eventually selected
as the consensus meeting. In particular, if a client in the legacy clients 108-B has declined
the meeting request for the meeting that will eventually become the consensus meeting, an
invite for that meeting may need to be resent to any such clients that had declined the
meeting request so that the eventual consensus meeting could be added to the legacy clients
calendar view.

[0026] In some embodiments, the instance messages 106 may be specially marked to
indicate to the users at the legacy clients 108-B that they are alternative meeting requests.
For example, the subject fields of the instance messages 106 may include an indication that
several alternatives for a single meeting are being sent and that a user should vote for a
meeting alternative by accepting one meeting request and declining others for which they
do not wish to vote. In these embodiments, once a consensus meeting is chosen by a meeting
organizer, a cancellation may be sent for all of the meeting requests represented by the
instance messages 106 (or at least any of the meeting requests that were accepted) to clear
the path for a new meeting request to be sent in the messages 122 which would not include
the specially marked notation indicating that it is an alternative meeting request. Instead, in
some embodiments, the messages 122 may include meeting requests which are invites to
the consensus meeting and which indicate that the consensus meeting was selected from the
alternatives. In this way, users at the legacy clients can be alerted that a consensus has been
reached and alerted to the selected consensus meeting.

[0027] Various additional details are now illustrated. Often the server 102 will not know
which clients are legacy clients and which clients are modern clients. As such, embodiments
may be implemented where the master message 104 and all of the instance messages 106
are all sent to all modern clients 108-A and all legacy clients 108-B. The modern clients
102-A can discard the instance messages and use the master message 104, while the legacy
clients 108-B can discard the master message 104 and use the instance message 106.

[0028] However, in some embodiments, the master message 104 may include the
alternatives A1, A2 and A3 by including pointers to the actual values contained in the
instance messages 106-1, 106-2 and 106-3 respectively. In this case, the modern clients
would obtain the default values \(D_1, D_2, D_3, \text{ and } D_4\) from the master message 104 while following the pointers to the instance messages 106-1, 106-2 and 106-3 to obtain the alternative values \(A_1, A_2, \text{ and } A_3\) respectively.

[0029] Messages 120 and 122 may be sent to all clients as well for some embodiments. However, in other embodiments, the server 102 may be able to determine by the responses 114 and 118 (or through other protocol communications) what type of client is being communicated with and thereafter selectively send only the correct type of message to the clients based on legacy or modern status of the clients.

[0030] To send the messages 104, 106, 120, and 122 and to receive the messages 114 and 188, the server may use mailboxes 109-A and 109-B. In particular, messages can be stored at the mailboxes 109-A and 109-B by the server 102. The clients can then access the mailboxes 109-A and 109-B to retrieve the messages as appropriate. In some embodiments, the messages may be stored as email messages.

[0031] Further, while the examples illustrated herein show communication directly between the server 102 and the clients 108, embodiments may be implemented where intervening servers can be used as well. For example, instead of communicating directly with the clients 108, the server 102 may communicate with other servers which store the mailboxes 109-A and 109-B instead of the mailboxes being stored at the server 102. Notably, the other servers could be either legacy servers or modern servers. In some such embodiments, the server 102 will send all messages 104 and 106 (and potentially 120 and 122) to all clients 108 irrespective of whether or not the clients are legacy clients or modern clients. Further, the messages 104 and 106 (and potentially 120 and 122) will be sent in a format that is able to be handled by any intermediate server irrespective of whether or not the server is a modern server or a legacy server.

[0032] Details are now illustrated regarding embodiments that can be used for creation of the instance messages 106 from the master message 104.

[0033] Some embodiments herein may be implemented using a master message and a set of instance messages for a series of messages. The master message stores all of the default values for the series of messages. The instance messages store any exceptions to the default values. It may be desirable to apply the default values to the instance messages for any values that are not exception values. This may be particularly true when a default value is updated and that update needs to be propagated to the instance messages. Thus, embodiments may apply the same operation to a number of distinct items, in this case,
messages. In some embodiments, the messages may be calendar items, and the series of
messages may be a series of recurring calendar items.

[0034] Referring now to Figure 2A, an example is illustrated. Figure 2 illustrates a series
of messages. The series 100 of messages includes a master message 104 and a set of
instance messages 106-1, 106-2 and 106-3. The master message 104 includes a plurality of
default values D1, D2, D3, D4, D5, and D6. In the example where the series 100 of messages
are calendar items, these values might include values defining dates, times, meeting
attendees, locations, etc.

[0035] Figure 2A further illustrates the instance messages 106-1, 106-2 and 106-3. While three instance messages are shown, it should be appreciated that any appropriate
number of messages may be used. These instance messages include exceptions to the default
values in the master message 104. For example, instance message 106-1 is shown with an
to the instance messages 106-1, 106-2 and 106-3. Thus, in the illustrated example, the default values D2 through D5 may need to be
applied to the instance messages 106-1, 106-2 and 106-3. Various operations may be
performed to apply the default values D2 through D5 to the instance messages.

[0037] Thus, in this example, the same operation(s) need(s) to be applied to a number
of distinct items, in this example, messages. Embodiments may have a command queue 105
in which the command to perform an operation is logged. In this example shown in Figure
2A, the command queue 105 is included in the master message 104. Thus, in the example,
for non-pattern recurrence embodiments (as discussed in more detail below), there is a
command queue 105 on each series master (e.g., master message 104) which is used to store
any series level updates. These need to be applied, in the order they appear on the queue, to
each instance message 106 to facilitate interoperability with legacy clients. While the
examples here illustrate the command queue 105 on each master message 104, in other
embodiments, the command queue 105 may be stored in other locations and associated with
the master message 104.

[0038] In the illustrated example, there are two mechanisms configured to apply update
commands from the command queue 105 to the instance messages, namely an in-line tool
110 and a background service 112 which will apply the command. On a series update, a
command is queued up by the in-line tool 110 which tries to apply the command to each
individual instance message 106. The in-line tool 110 may be, for example, an application
programming interface (API) on a server 102. For example, the server 102 may be a
calendar system such as the calendaring system available in Exchange Server available
from Microsoft Corporation of Redmond, Washington.

[0039] A call to the in-line tool 110 may be terminated due to system failure, operating
errors, or for some other reason, in between when the call to the in-line tool 110 is made and
when updates have been applied to instance messages. However, as noted, embodiments
may include a background service 112 which obtains commands from the command queue
105 and applies these commands to the instance messages 106 in concert with the in-line
tool 110. As the background service 112 is running independently of the in-line tool 110,
there could be a race condition with the inline tool 110. Additionally or alternatively,
resources may be wasted when the background service 112 first checks to see if a particular
command has already been applied to each instance message 106. To optimize on both of
these, embodiments may be configured to have the background service 112 apply
commands from the command queue 105 to the instance messages 106-1, 106-2 and 106-3
in reverse order with respect to the order used by the in-line tool 110. For example, if the
instance messages 106-1, 106-2 and 106-3 are ordered, e.g. ordered by their start times, then
the inline tool is configured to update 106-1, then 106-2, and then 106-3.

Contemporaneously, the background service is configured to start with 106-3, then 106-2,
and then 106-1. For example, Figure 2B illustrates an example where the in-line tool 110
applies value D2 to the instance message 106-1 while the background service 112 applies
the value D2 to the instance message 106-3.

[0040] Figure 2C illustrates an example where the in-line tool 110 has stopped applying
updates for some reason. For example, perhaps the in-line tool 110 has encountered an error.
In this example, the background service 112 applies the value D2 to the instance message
106-2. Since the command for applying the value D2 to the instance messages 106 has
completed, the background service 112 starts executing the command(s) for applying the
value D3 to the instance messages 106. In Figure 2C, the value D3 is applied to the instance
message 106-3 by the background service 112. Figure 2D illustrates that the value D3 is
then applied to the instance message 106-2 by the background service 112.

[0041] Propagation of values from the master message 104 to the instance messages
106-1, 106-2, and 106-3 may be performed in a number of different fashions. For example,
in the examples illustrated in Figures 1A, IB, 1C and ID default values are propagated in a
first instance when the instance messages have no pre-existing default values or
corresponding exception values. Thus, Figures 1A, IB, 1C and ID illustrate examples where
exception values El-1, El-2, and El-3 exist superseding the default value Dl. Other than
the exception values El-1, El-2, and El-3, the instance messages 106-1, 106-2, and 106-3
do not include, initially, any of the other default values D2, D3, D4, D5, and D6. Figures
1A, IB, 1C and ID illustrate initial application of the default values D2, D3, D4, D5, and
D6. Figures 1A, IB, 1C and ID illustrate an example where default values are added one
default value at a time to the instance messages.

[0042] Alternatively, as illustrated in Figure 3, when initially applying default values,
embodiments could add all appropriate defaults to an instance message (i.e., default values
for which there is not a superseding exception value) and then move to next instance
message. In the example illustrated in Figure 3, the in-line tool 110 applies default values
D2, D3, D4, D5, and D6 to the instance message 106-1 while the background service 112
applies the default values D2, D3, D4, D5, and D6 to the instance message 106-3.

[0043] In yet an alternative embodiment, the default values DI, D1, D2, D3, D4, D5, and
D6 are applied to the instance messages 106-1, 106-2, and 106-3, as appropriate, when those
messages are created and exception values El-1, El-2, and El-3 can be applied to instance
messages 106-1, 106-2, and 106-3 respectively later. Alternatively, the exception values El-
1, El-2, and El-3 can be applied to instance messages 106-1, 106-2, and 106-3 respectively,
while the default values D2, D3, D4, D5, and D6 are applied during the creation process of
the instance messages 106-1, 106-2, and 106-3.

[0044] Once default values have been applied to the instance messages 106-1, 106-2,
and 106-3 there may be a need to update a default value that should be applied to all
messages. For example, Figure 4A illustrates that default value D3 is updated to D3’ in the
master message 104. This change is propagated to the instance messages in a fashion as
illustrated above. For example, in Figure 4A, the in-line tool 110 (from Figure 2A) is used
to update the instance message 106-1 while the background service 112 is used to update
the instance message 106-3. Figure 4B illustrates completion of updating all instance
messages 106-1, 106-2 and 106-3 by using the in-line tool 110 and/or the background
service 112.

[0045] Notably, however, updating the default value D3 to D3’ does not result in an
overwrite of the exception values El-1, El-2, and El-3 as would normally occur in some
legacy systems.
In some embodiments, the messages are email messages. For example, in some embodiments, a string of emails may exist. In some legacy systems, hashtags for a string of emails, or social media "likes" of the string of emails may be able to be added to the entire string. However, to remove a "like" or a hashtag from an individual message in the string, embodiments can create an exception that indicates the removal of the "like" or hashtag. The exception can be propagated as appropriate to an instance message while maintaining other default values.

The following discussion now refers to a number of methods and method acts that may be performed. Although the method acts may be discussed in a certain order or illustrated in a flow chart as occurring in a particular order, no particular ordering is required unless specifically stated, or required because an act is dependent on another act being completed prior to the act being performed.

Referring now to Figure 5, a method 500 is illustrated. The method 500 may be practiced in a computing environment and includes acts for facilitating using a set of alternative events with both modern and legacy systems. The method 500 includes identifying a master message including a set of alternative events formatted for a modern system, wherein the master message comprises default information for all events in the set of alternative events as well as exception information representing exceptions to the default information for individual events in the set of alternative events (act 502).

The method 500 further includes identifying a plurality of instance messages, wherein each of the instance messages comprises default information from the master message and exception information for a given event in the set of alternative events (504).

The method 500 further includes sending the master message and the instance messages to other systems without regard to whether the other systems are modern or legacy systems so as to allow modern systems to select event alternatives using the master message and to allow legacy systems to select event alternatives using the instance messages (506).

The method 500 may be practiced where the master message is usable by modern systems to allow users at modern clients to select an alternative.

The method 500 may be practiced where the instance messages are meeting invites that allow legacy clients to select an alternative by accepting one of the instance messages and declining the rest of the instance messages.

The method 500 further includes receiving from modern clients votes for one or more of the events in the plurality of events; receiving from legacy clients one or more acceptances and one or more declines for one or more of the events in the plurality of events;
and using the votes, the acceptances and the declines creating an identification of a consensus of one or more of the alternative events. In some such embodiments, the method 500 may further include sending an event request for the identified consensus to both modern and legacy clients. Alternatively or additionally, in some such embodiments, the method 500 may further include sending out a cancellation for all of the events in the set of events except one or more selected consensus events selected by an organizer to the legacy clients. Alternatively or additionally, in some such embodiments, the method 500 may further include sending out a cancellation for all of the events in the set of events and sending out one or more consensus event invites selected by an organizer to the legacy clients.

[0054] The method 500 may be practiced where the exception information is included in the master message by using pointers to the instance messages.

[0055] The method 500 further includes creating the instance messages by using the master message and propagating information from the master message to the instance messages.

[0056] Further, the methods may be practiced by a computer system including one or more processors and computer-readable media such as computer memory. In particular, the computer memory may store computer-executable instructions that when executed by one or more processors cause various functions to be performed, such as the acts recited in the embodiments.

[0057] Embodiments of the present invention may comprise or utilize a special purpose or general-purpose computer including computer hardware, as discussed in greater detail below. Embodiments within the scope of the present invention also include physical and other computer-readable media for carrying or storing computer-executable instructions and/or data structures. Such computer-readable media can be any available media that can be accessed by a general purpose or special purpose computer system. Computer-readable media that store computer-executable instructions are physical storage media. Computer-readable media that carry computer-executable instructions are transmission media. Thus, by way of example, and not limitation, embodiments of the invention can comprise at least two distinctly different kinds of computer-readable media: physical computer-readable storage media and transmission computer-readable media.

[0058] Physical computer-readable storage media includes RAM, ROM, EEPROM, CD-ROM or other optical disk storage (such as CDs, DVDs, etc), magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store desired
program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer.

[0059] A "network" is defined as one or more data links that enable the transport of electronic data between computer systems and/or modules and/or other electronic devices. When information is transferred or provided over a network or another communications connection (either hardwired, wireless, or a combination of hardwired or wireless) to a computer, the computer properly views the connection as a transmission medium. Transmissions media can include a network and/or data links which can be used to carry out desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer. Combinations of the above are also included within the scope of computer-readable media.

[0060] Further, upon reaching various computer system components, program code means in the form of computer-executable instructions or data structures can be transferred automatically from transmission computer-readable media to physical computer-readable storage media (or vice versa). For example, computer-executable instructions or data structures received over a network or data link can be buffered in RAM within a network interface module (e.g., a "NIC"), and then eventually transferred to computer system RAM and/or to less volatile computer-readable physical storage media at a computer system. Thus, computer-readable physical storage media can be included in computer system components that also (or even primarily) utilize transmission media.

[0061] Computer-executable instructions comprise, for example, instructions and data which cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. The computer-executable instructions may be, for example, binaries, intermediate format instructions such as assembly language, or even source code. Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the described features or acts described above. Rather, the described features and acts are disclosed as example forms of implementing the claims.

[0062] Those skilled in the art will appreciate that the invention may be practiced in network computing environments with many types of computer system configurations, including, personal computers, desktop computers, laptop computers, message processors, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, mobile
telephones, PDAs, pagers, routers, switches, and the like. The invention may also be practiced in distributed system environments where local and remote computer systems, which are linked (either by hardwired data links, wireless data links, or by a combination of hardwired and wireless data links) through a network, both perform tasks. In a distributed system environment, program modules may be located in both local and remote memory storage devices.

[0063] Alternatively, or in addition, the functionally described herein can be performed, at least in part, by one or more hardware logic components. For example, and without limitation, illustrative types of hardware logic components that can be used include Field-programmable Gate Arrays (FPGAs), Program-specific Integrated Circuits (ASICs), Program-specific Standard Products (ASSPs), System-on-a-chip systems (SOCs), Complex Programmable Logic Devices (CPLDs), etc.

[0064] The present invention may be embodied in other specific forms without departing from its spirit or characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.
CLAIMS

1. A computing system for facilitating using a set of alternative events with both modern and legacy systems, the computing system comprising:
   one or more processors; and
   one or more computer-readable media, wherein the one or more computer-readable media comprise computer-executable instructions that when executed by the one or more processors, cause the computing system to perform the following:
   identify a master message including a set of alternative events formatted for a modern system, wherein the master message comprises default information for all events in the set of alternative events as well as exception information representing exceptions to the default information for individual events in the set of alternative events;
   identify a plurality of instance messages, wherein each of the instance messages comprises default information from the master message and exception information for a given event in the set of alternative events; and
   send the master message and the instance messages to other systems without regard to whether the other systems are modern or legacy systems so as to allow modern systems to select event alternatives using the master message and to allow legacy systems to select event alternatives using the instance messages.

2. The computing system of claim 1, wherein the master message is usable by modern systems to allow users at modern clients to select an alternative.

3. The computing system of claim 1, wherein the instance messages are meeting invites that allow legacy clients to select an alternative by accepting one of the instance messages and declining the rest of the instance messages.

4. The computing system of claim 1, wherein the one or more computer-readable media comprise computer-executable instructions that when executed by the one or more processors cause the computing system to perform the following:
   receive from modern clients' votes for one or more of the events in the plurality of events;
   receive from legacy clients one or more acceptances and one or more declines for one or more of the events in the plurality of events; and
   use the votes, the acceptances and the declines creating an identification of a consensus of one or more of the alternative events.
5. The computing system of claim 4, wherein the one or more computer-readable media comprise computer-executable instructions that when executed by the one or more processors cause the computing system to send an event request for the identified consensus to both modern and legacy clients.

6. The computing system of claim 4, wherein the one or more computer-readable media comprise computer-executable instructions that when executed by the one or more processors cause the computing system to send out a cancellation for all of the events in the set of events except one or more selected consensus events selected by an organizer to the legacy clients.

7. The computing system of claim 4, wherein the one or more computer-readable media comprise computer-executable instructions that when executed by the one or more processors cause the computing system to send out a cancellation for all of the events in the set of events and send out one or more consensus event invites selected by an organizer to the legacy clients.

8. A computer-implemented method of facilitating using a set of alternative events with both modern and legacy systems, the computer-implemented method being performed by one or more processors executing computer executable instructions for the computer-implemented method, and the computer-implemented method comprising:

   identifying a master message including a set of alternative events formatted for a modern system, wherein the master message comprises default information for all events in the set of alternative events as well as exception information representing exceptions to the default information for individual events in the set of alternative of events;

   identifying a plurality of instance messages, wherein each of the instance messages comprises default information from the master message and exception information for a given event in the set of alternative events; and

   sending the master message and the instance messages to other systems without regard to whether the other systems are modern or legacy systems so as to allow modern systems to select event alternatives using the master message and to allow legacy systems to select event alternatives using the instance messages.

9. The computer-implemented method of claim 8, further comprising:

   receiving from modern clients votes for one or more of the events in the plurality of events;
receiving from legacy clients one or more acceptances and one or more declines for one or more of the events in the plurality of events; and
using the votes, the acceptances and the declines creating an identification of a consensus of one or more of the alternative events.
10. The computer-implemented method of claim 9, further comprising sending an event request for the identified consensus to both modern and legacy clients.
### Figure 3

<table>
<thead>
<tr>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D3'</th>
<th>D4</th>
<th>D4</th>
<th>D5</th>
<th>D5</th>
<th>D6</th>
<th>D6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Figure 4A

<table>
<thead>
<tr>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D3'</th>
<th>D4</th>
<th>D4</th>
<th>D5</th>
<th>D5</th>
<th>D6</th>
<th>D6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Figure 4B

<table>
<thead>
<tr>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D3'</th>
<th>D4</th>
<th>D4</th>
<th>D5</th>
<th>D5</th>
<th>D6</th>
<th>D6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Identify a Master Message Including a Set of Alternative Events Formatted For a Modern System, Wherein the Master Message Comprises Default Information for All Events in the Set of Alternative Events as well as Exception Information Representing Exceptions to the Default Information for Individual Events in the Set of Alternative Events.

Identify a Plurality of Instance Messages, Wherein Each of the Instance Messages Comprises Default Information from the Master Message and Exception Information for a Given Event in the Set of Alternative Events.

Send the Master Message and the Instance Messages to Other Systems Without Regard to Whether the Other Systems Are Modern or Legacy Systems So As to Allow Modern Systems to Select Event Alternatives Using the Master Message and to Allow Legacy Systems to Select Event Alternatives Using the Instance Messages.

Figure 5
### A. CLASSIFICATION OF SUBJECT MATTER

INV. G06Q10/10

ADD.

According to International Patent Classification (IPC) or to both national classification and IPC.

### B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched.

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

### C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>US 2013/067024 Al (VASTERS CLEMENS FRIEDRICH [US]) 14 March 2013 (2013-03-14) page 1, paragraph 7</td>
<td>1-10</td>
</tr>
<tr>
<td></td>
<td>page 2, paragraph 19 - page 9, paragraph 109 figures</td>
<td></td>
</tr>
<tr>
<td></td>
<td>page 2, paragraph 30 - page 5, paragraph 58 figures</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>Wo 99/22324 Al (MICROSOFT CORP [US]) 6 May 1999 (1999-05-06) page 37, line 11 - page 39, line 6</td>
<td>1-10</td>
</tr>
</tbody>
</table>

[X] Further documents are listed in the continuation of Box C.  
[X] See patent family annex.

**Special categories of cited documents:**

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier application or patent but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) one of which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

**Later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention:**

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- "A" document member of the same patent family

**Date of the actual completion of the international search:**  
1 November 2016

**Date of mailing of the international search report:**  
10/11/2016

**Name and mailing address of the ISA:**

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel.: +31-70 340-2040, Fax: +31-70 340-3016

**Authorized officer:**

Rachkov, Vassi l
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>US 2014/229560 Al (GRAY MICHAEL HUNTER [US]) 14 August 2014 (2014-08-14) page 9, paragraph 105-106 page 12, paragraph 144 page 14, paragraph 157-158</td>
<td>1-10</td>
</tr>
<tr>
<td>Patent document cited in search report</td>
<td>Publication date</td>
<td>Patent family member(s)</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>US 2013067024 A1</td>
<td>14-03-2013</td>
<td>CN 103051667 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 2756475 A2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2014528126 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 20140072044 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2013067024 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO 2013039798 A2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2013212597 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 1025526 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 4575591 B2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2001521253 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO 9922324 A1</td>
</tr>
<tr>
<td>US 2014229560 A1</td>
<td>14-08-2014</td>
<td>NONE</td>
</tr>
<tr>
<td>US 2009248474 A1</td>
<td>01-10-2009</td>
<td>NONE</td>
</tr>
<tr>
<td>US 2013290058 A1</td>
<td>31-10-2013</td>
<td>NONE</td>
</tr>
</tbody>
</table>