(51) International Patent Classification:
G06F 9/44 (2006.01) G06F 3/048 (2006.01)
G06F 3/14 (2006.01) G06F 11/32 (2006.01)

(21) International Application Number:
PCT/US2012/057446

(22) International Filing Date:
27 September 2012 (27.09.2012)

(25) Filing Language: English

(26) Publication Language: English


(72) Inventors: SHERSHHEVSKY, Olga; Shabazi 19, 56100 Yehud (IL). ARVATS, Efrat; Shabazi 26, 56100 Yehud (IL). EISENBERG, Kobi; Shabazi 19, 56100 Yehud (IL).


Declarations under Rule 4.17:
— as to the identity of the inventor (Rule 4.17(i))

Published:
— with international search report (Art. 21(3))

(54) Title: CAPTURING AN APPLICATION STATE IN A CONVERSATION

FIG. 8

ACCESS AN ELECTRONIC DOCUMENT REPRESENTATION OF A USER INTERFACE AT A TIME CORRESPONDING TO A POST TO THE CONVERSATION, THE USER INTERFACE BEING OF AN APPLICATION ASSOCIATED WITH THE CONVERSATION

PROCESS THE REPRESENTATION TO GENERATE AN IMAGE REPRESENTATIVE OF A STATE OF THE APPLICATION AT THE TIME

CAUSE THE IMAGE OR A REPRESENTATION THEREOF TO BE DISPLAYED IN THE POST TO REFLECT THE STATE OF THE APPLICATION AT SELECTED POINT ALONG A TIMELINE DEFINED BY THE CONVERSATION

(57) Abstract: The state of an application is captured in a conversation by accessing an electronic document representation of a user interface at a time corresponding to a post to the conversation. The user interface is of an application associated with the conversation. The representation is processed to generate an image representative of a state of the application at the time. The image or a representation thereof is caused to be displayed in the post to reflect the state of the application at selected point along a timeline defined by the conversation.
CAPTURING AN APPLICATION STATE IN A CONVERSATION

BACKGROUND

[0001] Tools such as e-mail and chat enable participants to engage in conversations and collaborate on given topics. The context of such conversations is discerned by the participants reading the conversation. Initially, the context may be defined by a subject line. As an e-mail or chat thread grows with static files being added as attachments, the current topic of discussion can change over time. As a result, new and existing participants can find it cumbersome to discern the current context making it more difficult to take actions called for by the conversation.

DRAWINGS

[0002] Figs. 1-4 depict screen views of a user interface displaying a conversation where the state of an application is captured in the conversation according to examples.

[0003] Fig. 5 depicts an example environment in which various embodiments may be implemented.

[0004] Figs. 6 and 7 depict a system according to an example.

[0005] Fig. 8 is a flow diagram depicting steps taken to implement an example.
DETAILED DESCRIPTION

[0006] **INTRODUCTION:** It is often the case that enterprise software applications, especially in IT organizations, are complex. The applications can require significant training and knowhow in order for a user to extract relevant information or perform specific actions of a given work flow. The information that a user consumes and the actions that are performed are often dynamic and defined by the context of that work flow. Before an action is taken, a user may collaborate with others to obtain perspective, guidance, or even permission. Conventionally, users, when faced with action items in a conversation, access and manipulate information in applications other than the tool used to collaborate.

[0007] Various embodiments described below enable collaboration participants to consume the relevant application information from within a collaboration tool. As described in more detail below, this relevant information includes the states of the application or applications that are the subject of the conversation. A conversation, as used herein, is an electronic interaction between participants concerning an evolving topic in the context of an application. Each participant can be presented with collaboration content that includes a conversation timeline made up of posts of the various participants.

[0008] An example of capturing an application state in a conversation includes, at a time corresponding to a post to the conversation, accessing an electronic document representation of a user interface of an application associated with the conversation. The electronic document representation is processed to generate an image representative of a state of the application at the time corresponding to the post. The image or a representation thereof is caused to be displayed in the post to reflect the state of the application at selected point along a timeline defined by the conversation.

[0009] Electronic document representation of a user interface can include markup language representations and DOMs (Document Object Models) generated from markup language representations or any other electronic document containing content that when processed can be used to cause a
display of a user interface. In an example, the application may be a web or
a cloud application. A client interacts with the application by sending
request. Responses to those requests can include a markup language
document that is processed and displayed. In processing the markup
language document, the client may generate a DOM which serves as an API
(application programming interface) providing a structural representation of
the markup language document allowing its content and visual presentation
to be modified using a scripting language. Here, the clients can save the
markup language or DOM document for later use in generating the image for
inclusion in the conversation post. This image, for example, can be
generated by using an HTML canvas element. The canvas element is a
container for graphics drawn by a script processing the electronic document
representation. The script, for example, produce an image that includes all
elements of the electronic document representation, or the script may take,
as input, identifiers for selected elements. A DOM organizes the elements of
a user interface (from the markup language document), in a hierarchical tree
structure allowing particular elements to be selected and used by the script
in generating the image.

[0010] The following description is broken into sections. The first, labeled
"Illustrative Examples," presents examples of application states being
captured in a conversation. The second section, labeled "Environment,"
describes an example environment in which various embodiments may be
implemented. The third section, labeled "Components," describes examples
of various physical and logical components for implementing various
embodiments. The fourth section, labeled as "Operation," describes steps
taken to implement various embodiments.

[0011] Illustrative Examples: Figs. 1-4 depict an example where
application states are captured in a conversation. Fig. 1 depicts sample
corresponds conversation made up of posts along a timeline. Figs. 2-4 each depict an
example of how an application state can be captured in a post to that
corresponds. Starting with Fig. 1, user interface 10 is of a collaboration tool
and is divided into three main sections 12, 14, and 16. Section 12 provides
an area to display a list of collaborations. Each collaboration can involve a
different topic and can include different participants. In Fig. 1, collaboration
18 is selected. Collaboration 18 relates to an application bug being tracked
by a defect management tool provided by an application identified as
“Quality Center.” The conversation can be said to be associated with the
Quality Center application as well as the application suffering from the defect
referred to here as “Banking App.” The conversation may be associated
with the Banking App via an application testing tool used to identify the
defect. As will be expanded on below, the various states of the defect and
the Banking App as reported by the defect management tool and the testing
tool can be captured

[0012] As a result of selecting collaboration 18, collaboration content for
collaboration 18 is displayed in sections 14 and 16. That collaboration
content includes a collaboration timeline 20 displayed in section 16 and facet
22 of the bug reporting tool in section 16. Collaboration timeline 20 as
shown has grown over time to include posts 24-34. Posts 24-34 of
conversation 18 can be said to define timeline 20. Here timeline 20 starts
with post 24 and ends with post 34. As depicted, timeline 20 includes posts
24, 32, and 34 each corresponding to a different screenshot of a given
application. Here, posts 24 and 32 correspond to screenshots capturing the
state of Banking App as reported by the testing tool at different points along
timeline 20. Post 34 corresponds to a screenshot capturing a state of the
defect tracking tool at another point along collaboration timeline 20.

[0013] The term screenshot here refers to an image generated by
processing an electronic document representation of the user interface for
the defect management tool for post 34 and an electronic document
representation of the user interface for the testing tool for posts 24 and 32.
The electronic document representation may be obtained for or generated
from data obtained from the respective applications. Such data may be
maintained by or otherwise accessible from the device displaying interface
10. For example, the conversation here is associated with a particular
application defect that can be distinguished using an identifier within the
defect management tool. That defect in turn can be associated with a particular test within the test tool. Thus by using the identifier for the defect, the electronic document models can be identified on the client or requested for the respective tools.

[0014] The images, for example, can be generated in response to a capture event. The capture event may be triggered by a user selecting a control such as control 36, or it may be triggered by another event such as, in Fig. 1, the creation or closing of a defect in the defect management tool. A capture event may lead to generating a single image capturing the states of one or more applications. A capture event may lead to generating multiple images capturing the states of multiple applications. In the example of Fig. 1, the closing of the defect in the defect management tool may have served as a capture event for generating an image capturing the state of the Banking App for post 32 and another image capturing the state of the defect management tool for post 34.

[0015] The image itself can be included in a given post. However, in the example of Fig. 1, representations of such images are included in posts 24, 32, and 34 in the form of links 40, 42, and 44 shown in Figs. 2-4 respectively. In Fig. 2, a user has selected link 40 of post 24 causing a pop-up display of image 42. Here image 42 captures a state of the Banking App captured through the testing tool and indicates an unexpected login error. In Fig. 3, the user has selected link 44 of post 32 causing a pop-up display of image 46. Here image 46 captures a later state of the Banking App captured through the testing tool. That state indicates a successful log-in. Finally, in Fig. 4, the user has selected link 48 causing a pop-up display of image 50. Image 50 captures a state of the defect management tool showing a history of the defect.

[0016] Environment: Fig. 5 depicts an environment 54 in which various embodiments may be implemented. Environment 54 is shown to include application services 56, state capture system 58, and client devices 60. Application services 56 each represent a computing device or combination of computing devices configured to serve an application to client devices 60.
Examples can include enterprise and consumer web and cloud applications provided through service oriented architectures.

[0017] State capture system 58 represents a computing device or combination of computing devices configured to capture the state of an application in a conversation associated with that application. System 58 is described in more detail below with respect to Figs. 6 and 7, but in general system 58 is configured to generate an image reflecting the state of that application and then cause that image or a representation thereof to be included in a post to that conversation. That post then is reflective of the application state at a given point along a timeline defined by post to that conversation.

[0018] Client devices 60 each represent a computing device configured to interact with application services 56 and state capture system 58. Such interaction may be through a browser or other application executing on a given client device 60. The applications can include the applications that are the subject of a conversation as well as a collaboration tool for managing the conversation. Link 40 represents generally one or more of a cable, wireless, fiber optic, or remote connections via a telecommunication link, an infrared link, a radio frequency link, or any other connectors or systems that provide electronic communication. Link 40 may include, at least in part, an intranet, the Internet, or a combination of both. Link 40 may also include intermediate proxies, routers, switches, load balancers, and the like.

[0019] COMPONENTS: Fig. 6 depicts components of system 58 for capturing an application state in a conversation. In Fig. 6 various components are identified as engines 64-70. In describing engines 64-70, focus will be on each engine's designated function. However, the term engine, as used herein, refers to a combination of hardware and programming configured to perform a designated function. As is illustrated later with respect to Fig. 7, the hardware of each engine, for example, may include a processor and a memory, while the programming is code stored on that memory and executable by the processor to perform the designated function.
[0020] In the example of Fig. 6 system 58 includes snapshot engine 64, interface engine 66, image engine 68, and presentation engine 70. Snapshot engine 64 is configured to detect a snapshot event with respect to a conversation associated with an application. A snapshot event is an event indicating a time at which the state of an application is to be captured in the conversation. A snapshot event can be triggered manually by a user interacting with a user interface control such as control 36 of Figs. 1-4. A capture event may be triggered based on the occurrence of another event. For example, where a conversation is associated with item being managed by an application, a change in status of that item may trigger the capture event. Referring to Figs. 1-4, a change in status of the defect being managed by the defect management tool may trigger a capture event.

[0021] Interface engine 66 is configured to, in response to the snapshot engine 66 detecting a snapshot event, access an electronic document representation of the application's user interface. Again, the application is an application associated with the conversation. The electronic document representation can be accessed in a number of fashions. Where the representation is a markup language representation or a DOM created from that markup language document, the representation may be maintained locally by a client device used to interact with the associated application. In another example, the conversation is known to be associated with a given application and perhaps a particular item managed by that application. Interface engine 66 may communicate a request (that, if available, identifies the application object) to the application. The application may then return a markup language representation of a current user interface that can be processed to create a DOM.

[0022] Image engine 68 is configured to generate an image corresponding to visual characteristics of selected elements identified from the electronic document representation. The generated image is thus representative of a state of the application at the time of the snapshot event. The selected elements identified may be all the elements of the electronic document representation that can be used to generated the image. Alternatively, the
particular elements may be selected to include only those relevant to the application object in question or those otherwise relevant to the context of the conversation. Such a selection can be made based on the nature of a capture event. For example, if the capture event is triggered by the closing of a defect in a defect management tool, the selected elements may include a history that would describe how the defect was cured. Image engine 68 may, for example, utilize an HTML canvas element to generate the image. In such a case, image engine 68 may include a script to define the content of the container defined by the canvas element. That script would utilize the selected elements from the electronic document representation.

[0023] Presentation engine 70 is configured to cause the image or a representation thereof to be displayed in a post to the conversation to reflect the state of the associated application at selected point along a timeline defined by the conversation. The resulting post can include the image alone or it may also include descriptive text. The resulting post can include the representation alone or the representation along with the descriptive text. The representation is a link that when selected causes a display of the image. The link may be generic or a thumbnail of the image it represents.

[0024] The conversation can include multiple posts in which the state of an associated image is captured. Thus, the conversation can be used to depict the evolving state of the application along a timeline. Where, for example, a conversation is on the topic of correcting an IT problem with respect to a given application, the conversation can be read later to learn if the solution reached might help with another problem. In this case, snapshot engine 64 would detect two snapshot events separated in time. Following the first snapshot event, engines 66-70 would perform their respective functions resulting in a first image being included in a first post. Following the second snapshot event, engines 66-70 would perform their respective functions again resulting in a second image being included in a second post. Assuming the state of the application had changed between snapshot events, the resulting images will differ and reflect the differing states
[0025] As noted with respect to the examples of Figs. 1-4, a conversation may be associated with multiple applications. Thus different posts to that conversation can include images or representations of images capturing the states for the different applications. Further, a single post can include one or more images or representations thereof capturing the states of the different applications. Here, the electronic document representation may be single DOM that represents user interfaces of the different applications allowing image engine 68 to process that DOM to generate the one or more images.

[0026] In foregoing discussion, various components were described as combinations of hardware and programming. Such components may be implemented in a number of fashions. Looking at Fig. 7, the programming may be processor executable instructions stored on tangible, non-transitory memory resource 74 and the hardware may include processing resource 76 for executing those instructions. Thus memory resource 74 can be said to store program instructions that when executed by processing resource 76 implement system 58 of Fig. 6.

[0027] Memory resource 74 represents generally any number of memory components capable of storing instructions that can be executed by processing resource. Such memory components being non-transitory computer readable media. Memory resource 74 may be integrated in a single device or distributed across devices. Likewise processing resource 76 represents any number of processors capable of executing instructions stored by memory resource. Processing resource 76 may be integrated in a single device or distributed across devices. Further, memory resource 74 may be fully or partially integrated in the same device as processing resource 76 or it may be separate but accessible to that device and processing resource 76. Thus, it is noted that system 58 may be implemented on a user or client device, on a server device or collection of servicer devices, or on a combination of the user device and the server device or devices.

[0028] In one example, the program instructions can be part of an installation package that when installed can be executed by processing resource 76 to
implement system 58. In this case, memory resource 74 may be a portable medium such as a CD, DVD, or flash drive or a memory maintained by a server from which the installation package can be downloaded and installed. In another example, the program instructions may be part of an application or applications already installed. Here, memory resource 74 can include integrated memory such as a hard drive, solid state drive, or the like.

[0029] In Fig. 7, the executable program instructions stored in memory resource 74 are depicted as snapshot module 78, interface module 80, image module 82, and presentation module 94. Snapshot module 78 represents program instructions that, when executed, cause processing resource 76 to implement snapshot engine 64 of Fig. 6. Interface module 80 represents program instructions that when executed cause the implementation of interface engine 66. Likewise, image module 82 and presentation module represent program instructions that when executed cause the implementation of image engine 69 and presentation engine 70, respectively.

[0030] **Operation:** Fig. 8 is a flow diagram of steps taken to implement a method for capturing an application state in a conversation. In discussing Fig. 8, reference may be made to the diagrams of Figs. 1-7 to provide contextual examples. Implementation, however, is not limited to those examples.

[0031] An electronic document representation of a user interface is accessed at a time corresponding to a post to the conversation (step 86). The user interface is of an application associated with the conversation. As explained above, such a representation may be a markup language representation or a DOM generated from the markup language representation. The representation may be maintained by a client or obtained from the application. Referring to Fig. 6, interface engine 66 may be responsible for implementing step 86 while interface module 80 of Fig. 7 represents the program instructions for doing so.

[0032] The representation is processed to generate an image representative of a state of the application at the time (step 88). Step 88 may be
accomplished by identifying desired elements of the user interface. The representation can be examined to identify visual characteristics of the desired elements where those characteristics are illustrative of the state of the application. The image is then generated according to those characteristics. The desired elements may be default elements, elements identified by a user, or elements selected automatically based on the nature of a capture event that triggered step 86. Referring again to Figs. 6 and 7, image engine 68 may be responsible for implementing step 88 while image module 82 represents the program instructions for doing so.

[0033] The image or a representation thereof is caused to be displayed in the post to reflect the state of the application at selected point along a timeline defined by the conversation (step 90). Referring again to Figs. 6 and 7, presentation engine 70 may be responsible for implementing step 88 while presentation module 84 represents the program instructions for doing so.

[0034] Conclusion: Figs. 1-4 depict examples of a user interface 10 in which an application state is captured in a conversation. The particular layouts and designs of user interface 10 depicted in Figs. 1-4 are examples only. Figs. 5-7 aid in depicting the architecture, functionality, and operation of various embodiments. In particular, Figs. 5-7 depict various physical and logical components. Various components are defined at least in part as programs or programming. Each such component, portion thereof, or various combinations thereof may represent in whole or in part a module, segment, or portion of code that comprises one or more executable instructions to implement any specified logical function(s). Each component or various combinations thereof may represent a circuit or a number of interconnected circuits to implement the specified logical function(s).

[0035] Embodiments can be realized in any computer-readable media for use by or in connection with an instruction execution system such as a computer/processor based system or an ASIC (Application Specific Integrated Circuit) or other system that can fetch or obtain the logic from computer-readable media and execute the instructions contained therein.
"Computer-readable media" can be any non-transitory media that can contain, store, or maintain programs and data for use by or in connection with the instruction execution system. Computer readable media can comprise any one of many physical media such as, for example, electronic, magnetic, optical, electromagnetic, or semiconductor media. More specific examples of suitable computer-readable media include, but are not limited to, a portable magnetic computer diskette such as floppy diskettes, hard drives, solid state drives, random access memory (RAM), read-only memory (ROM), erasable programmable read-only memory, flash drives, and portable compact discs.

[0036] Although the flow diagram of Fig. 8 shows a specific order of execution, the orders of execution may differ from that which is depicted. For example, the order of execution of two or more blocks or arrows may be scrambled relative to the order shown. Also, two or more blocks shown in succession may be executed concurrently or with partial concurrence. All such variations are within the scope of the present invention.

[0037] The present invention has been shown and described with reference to the foregoing exemplary embodiments. It is to be understood, however, that other forms, details and embodiments may be made without departing from the spirit and scope of the invention that is defined in the following claims.
CLAIMS

What is claimed is:

1. A system comprising a memory resource storing instruction executable by a processing resource to capture an application state in a conversation, the memory resource storing:
   an interface module to, when executed, access an electronic document representation of a user interface at a time corresponding to a post to the conversation, the user interface being of an application associated with the conversation;
   an image module to, when executed, process the representation to generate an image representative of a state of the application at the time;
   and
   a presentation module to, when executed, cause the image or a representation thereof to be displayed in the post to reflect the state of the application at selected point along a timeline defined by the conversation.

2. The system of Claim 1, wherein the image module is a module, when executed to:
   identify desired elements of the user interface;
   examine the representation to identify visual characteristics of the desired elements, the characteristics being illustrative of the state of the application; and
   generate the image according to those characteristics.

3. The system of Claim 1, wherein the electronic document representation is one of a markup language document representation and a DOM (document object model) representation.

4. The system of Claim 1, wherein:
   the conversation is associated with a plurality of applications;
the electronic document representation is one or more electronic
document representations of one or more user interfaces for the plurality of
applications;
the image module is a module, when executed, to process the one or
more representations to generate one or more images representative of
states of the plurality of applications at the time; and
the presentation module is a module, when executed, to cause the
one or more images or a representation thereof to be displayed in the post to
reflect the states of the plurality of applications at selected point along a
timeline defined by the conversation.

5. The system of Claim 1, wherein:
the memory resource stores a snapshot module to, when executed,
detect a snapshot event instigated via a user interface depicting the
conversation and not the user interface of the associated application, and
the time coincides with the snapshot event.

6. The system of Claim 1, wherein:
the electronic document representation is a first representation, the
state is a first state, and the image is a first image;
the interface module is a module, when executed, to access a second
electronic document representation of the user interface at a second time,
the second time corresponding to a second post to the conversation,
the image module is a module, when executed, to process the second
representation to generate a second image representative of a second state
of the application at the second time;
the presentation module is a module, when executed, to cause the
second image or a representation thereof to be displayed in the second post
to reflect the second state of the application such that the conversation
includes a plurality of posts depicting changing states of the application
along the timeline.
7. The system of Claim 1, further comprising the processing resource.

8. A computer readable medium having instructions that when executed trigger a processor to:
   examine, in response to a snapshot event, an electronic document representation of an application's user interface to identify visual characteristics of selected elements;
   generate a digital image according to the identified characteristics;
   cause the digital image or a representation thereof to be displayed in a post to a conversation associated with the application, the digital image reflecting a state of the application at the time of the capture event.

9. The medium of Claim 8 wherein the electronic document is one of a markup language document representation and a DOM (document object model) representation.

10. The medium of Claim 8, wherein the conversation is associated with a plurality of applications and the instructions that trigger the processor to examine, generate and cause include instructions that when executed trigger the processor to:
   examine one or more electronic document representations of one or more user interfaces for the plurality of applications to identify visual characteristics of selected elements;
   generate one or more digital images according to the identified characteristics; and
   cause the one or more digital images or representations thereof to be displayed in the post to the conversation associated with the plurality of applications, the one or more digital images reflecting states of the plurality of applications at the time of the capture event.
11. The medium of Claim 8, wherein the conversation is associated with a first application and a second application and the instructions that trigger the processor to examine, generate and cause include instructions that when executed trigger the processor to:
   examine, in response to a first snapshot event, a first electronic document representation of the first application’s user interface to identify first visual characteristics of selected first elements
   generate a first digital image according to the identified first characteristics;
   cause the first digital image or a representation thereof to be displayed in a first post to the conversation, the first digital image reflecting a state of the first application at the time of the first capture event.
   examine, in response to a second snapshot event, a second electronic document representation of the second application’s user interface to identify second visual characteristics of selected second elements
   generate a second digital image according to the identified second characteristics;
   cause the second digital image or a representation thereof to be displayed in a second post to the conversation, the second digital image reflecting a state of the second application at the time of the second capture event.

12. The medium of Claim 8, wherein the electronic document representation is a first representation examined at a first time, the snapshot event is a first snapshot event, the state is a first state, the visual characteristics are first visual characteristics, the digital image is a first digital image, the post is a first post, and the instructions that trigger the processor to examine, generate and cause include instructions that when executed trigger the processor to:
1. examine, in response to a second snapshot event, a second
electronic document representation of the application’s user interface to
identify second visual characteristics of the selected elements;
generate a second digital image according to the second identified
characteristics;
cause the second digital image or a representation thereof to be
displayed in a second post to the conversation, the second digital image
reflecting a second state of the application at the time of the second capture
event.

13. A system comprising a snapshot engine, an interface engine,
an image engine, and a presentation engine, wherein:
the snapshot engine is configured to detect snapshot events with
respect to a conversation associated with an application;
the interface engine is configured, in response to the snapshot engine
detecting a snapshot event, to access an electronic document
representation of the application’s user interface;
the image engine is configured to generate an image corresponding
to visual characteristics of selected elements identified from the electronic
document representation, the image representative of a state of the
application at the time of the snapshot event; and
the presentation engine is configured to cause the image or a
representation thereof to be displayed in a post to the conversation to reflect
the state of the application at selected point along a timeline defined by the
conversation.

14. The system of Claim 13, wherein:
the conversation is associated with a plurality of applications;
the interface engine is configured, in response to the snapshot engine
detecting a snapshot event, to access electronic document representations
of user interfaces of the plurality of applications;
the image module is configured to generate one or more corresponding to visual characteristics of selected elements identified from the electronic document representations, the one or more images representative of states of the plurality of applications at the time of the snapshot event; and the presentation module to cause the one or more images or representations thereof to be displayed in the post to the conversation to reflect the states of the plurality of applications at selected point along a timeline defined by the conversation.

15. The system of Claim 13, wherein:

the electronic document representation is a first representation accessed at a first time, the snapshot event is a first snapshot event, the state is a first state, the visual characteristics are first visual characteristics, the image is a first image, and the post is a first post; the interface engine is configured, in response to the snapshot engine detecting a second snapshot event, to access a second electronic document representation of the application’s user interface;

the image module is configured to generate a second image corresponding to second visual characteristics of the selected elements identified from the second electronic document representation, the second image representative of a second state of the application at the time of the second snapshot event; and the presentation engine is configured to cause the second image or a representation thereof to be displayed in a second post to the conversation to reflect the second state of the application such that the conversation includes a plurality of posts depicting changing states of the application along the timeline.
Bank App Login - User Interface

- User Name: user abcd
- Password: **********
- Login

SUCCESS
please wait while we access your records

FIG. 3
QC Defect Manager - User Interface

<table>
<thead>
<tr>
<th>defect ID</th>
<th>owner</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D13221</td>
<td>Ithai</td>
<td>no contacts suggested</td>
</tr>
<tr>
<td>D13224</td>
<td>Susan</td>
<td>login fails</td>
</tr>
</tbody>
</table>

T1: Incident Reported Login Failing in Bank App
T2: Defect Created and Assigned to Susan
T3: Defect Cause Identified as ....
T4: Defect Remedied by....
T5: Bank App Tested - Login Functional

FIG. 4
FIG. 8

ACCESS AN ELECTRONIC DOCUMENT REPRESENTATION OF A USER INTERFACE AT A TIME CORRESPONDING TO A POST TO THE CONVERSATION, THE USER INTERFACE BEING OF AN APPLICATION ASSOCIATED WITH THE CONVERSATION

PROCESS THE REPRESENTATION TO GENERATE AN IMAGE REPRESENTATIVE OF A STATE OF THE APPLICATION AT THE TIME

CAUSE THE IMAGE OR A REPRESENTATION THEREOF TO BE DISPLAYED IN THE POST TO REFLECT THE STATE OF THE APPLICATION AT SELECTED POINT ALONG A TIMELINE DEFINED BY THE CONVERSATION
A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/44(2006.01)i, G06F 3/14(2006.01)i, G06F 3/048(2006.01)i, G06F 11/32(2006.01)i

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)
G06F 9/44; G06F 15/16; H04M 1/64; G06F 3/00; H04Q 11/00; G06K 9/00; H04L 12/16; G09G 5/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: conversation, application, state, timeline

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>A</td>
<td>US 2010-0138756 A1 (SAUND ERIC et al.) 03 June 2010 See abstract; paragraphs [0035]-[0123]; and figures 2-16.</td>
<td>1-15</td>
</tr>
<tr>
<td>A</td>
<td>US 2005-0021624 A1 (HERF MICHAEL et al.) 27 January 2005 See abstract; paragraphs [0042]-[0076]; and figures 1-5.</td>
<td>1-15</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C. 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) or which is cited to establish the publication date of 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

"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
"&" document member of the same patent family

Date of the actual completion of the international search: 30 April 2013 (30.04.2013)

Date of mailing of the international search report: 30 April 2013 (30.04.2013)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea
Facsimile No. 82-42-472-7140

Authorized officer

CHOI, Jeong Kwon
Telephone No. 82-42-481-8507

Form PCT/ISA/210 (second sheet) (July 2009)
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td>US 2010-0138756 A1</td>
<td>03.06.2010</td>
<td>EP 2192732 A2</td>
<td>02.06.2010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 2192732 A3</td>
<td>09.06.2010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2010-129093 A</td>
<td>10.06.2010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU 2011-200333 A1</td>
<td>17.02.2011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU 2011-200333 B2</td>
<td>08.03.2012</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BR P10410362A</td>
<td>04.07.2006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CA 2525939 A1</td>
<td>02.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 10258431 A</td>
<td>17.08.2011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 1997980 A</td>
<td>11.07.2007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 1997980 B</td>
<td>06.07.2011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 1997980 C0</td>
<td>11.07.2007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 1625476 A2</td>
<td>15.02.2006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 1625476 A4</td>
<td>27.02.2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2007-518146 A</td>
<td>05.07.2007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2007-518146 T</td>
<td>05.07.2007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 4647610 B2</td>
<td>09.03.2011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 10-1196271 B1</td>
<td>06.11.2012</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 10-2006-0026409 A</td>
<td>23.03.2006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 10-2011-0050750 A</td>
<td>16.05.2011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 7761507 B2</td>
<td>20.07.2010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO 2004-104758 A2</td>
<td>02.12.2004</td>
</tr>
<tr>
<td>US 2004-0228531 A1</td>
<td>18.11.2004</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>