(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 August 2014 (07.08.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/117387 A1

(51) International Patent Classification:
G06F 9/46 (2006.01)

(21) International Application Number:
PCT/CN2013/071264

(22) International Filing Date:
1 February 2013 (01.02.2013)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 11445 Compaq Center Drive West, Houston, Texas 77070 (US).

(72) Inventors; and
(71) Applicants *(for US only)*: **CHEN, Bo** [CN/CN]; No.20 Jia Feng Road, Pudong, Shanghai 200131 (CN). **LUAN, Jinfeng** [CN/CN]; No.20 Jia Feng Road, Pudong, Shanghai 200131 (CN). **SAARONI, Dror** [IL/IL]; Altalef st., Altalef st No.5, Shabazi St. No.26, 56100 Yehud (IL).

(74) Agent: **CHINA PATENT AGENT (HK) LTD.**; 22/F., Great Eagle Center, 23 Harbour Road, Wanchai, Hong Kong (CN).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) **Title**: TEST SCRIPT CREATION BASED ON ABSTRACT TEST USER CONTROLS

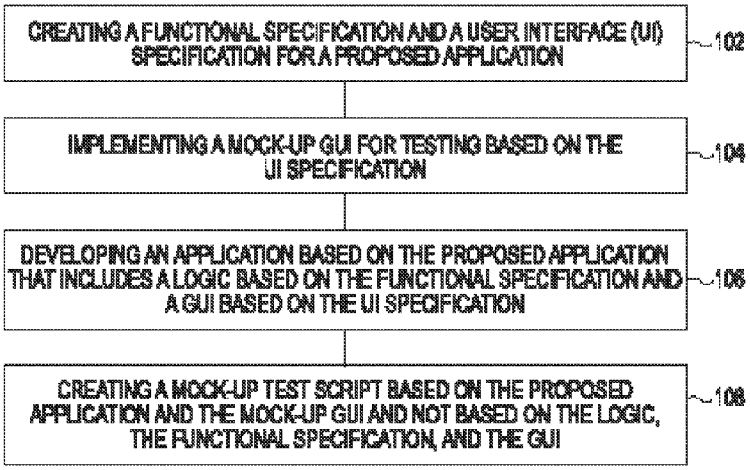

Fig. 1

(57) **Abstract**: The life cycle of an application can be shortened by initiating the creation of a mock-up test script before the development of an application is complete. The concurrent creation of the mock-up test script and the application can reduce a time associated with the life cycle of an application. Mock-up test script creation is based on the user interface (UI) specification and a mock-up graphical user interface (GUI) without business logic. Test script creation can include associating a functional specification and a UI specification with a proposed application and creating a mock-up GUI for testing based on the UI specification. Test script creation can include developing an application based on the proposed application that includes a logic based on the functional specification and a GUI based on the UI specification and creating a mock-up test script based on the mock-up GUI and not based on the logic, the functional specification, and the GUI.

WO 2014/117387 A1

TEST SCRIPT CREATION BASED ON ABSTRACT TEST USER CONTROLS

Background

[0001] The life cycle of an application can begin at a designing stage and can end at an execution stage of the application. An application is tested before it is distributed to customers. The testing of an application can find flaws in the application before the application is distributed to customers. Finding flaws and resolving the flaws can provide for a more enjoyable user experience than if the flaws are not resolved.

Brief Description of the Drawings

[0002] Figure 1 is a flow chart illustrating an example of a method for test script creation according to the present disclosure.

[0003] Figure 2 is a flow chart illustrating test script creation according to the present disclosure.

[0004] Figure 3 is a diagram illustrating an example of a mock up GUI for testing according to the present disclosure.

[0005] Figure 4 is a diagram illustrating an example of an application according to the present disclosure.

[0006] Figure 5 is a diagram illustrating an example of a computing system according to the present disclosure.

Detailed Description

[0007] The life cycle of an application can be shortened by initiating the creation of a mock-up test script before the development of an application is complete. The concurrent creation of the mock-up test script and the application can reduce a time associated with the life cycle of an application.

Shortening the time associated with the life cycle of an application can allow the application to be developed and tested in a shorter amount of time than if the developing and testing of an application is performed sequentially.

[0008] A life cycle of an application can define a number of stages associated with the creation and deployment of an application. For example, a life cycle can include a design stage, a development stage, a testing stage, and/or a deployment stage, among other stages. As used herein, a design stage can include the conception and the creation of a design for an application. As used herein, the conception and the creation of a design for an application can include the conception of a logic that is associated with an application and the conception of a design for a graphical user interface (GUI). A conceptualized logic and a conceptualized GUI is referred to herein as a functional specification and a user interface (UI) specification, respectively. The functional specification and the UI specification can be included in a proposed application.

[0009] The creation of a logic that is associated with an application can define what the application does. For example, a logic for a calculator application can perform a number of operations that are associated with mathematics. The logic for a calculator application can include addition logic, subtraction logic, multiplication logic, and/or division logic, among other types of logic. In various examples, logic for a login application can have logic associated with verifying that the information received from the user corresponds to the information of a user that has a number of rights and/or privileges.

[0010] The creation of a design for a GUI can include identifying a number of user controls that will be associated with a GUI. For example, a user control can be associated with a GUI when a user control is displayed on the GUI and/or when the GUI references the user control. Furthermore, the creation of a design for a GUI can also include defining a number of attributes that are associated with the number of user controls. For example, a button user control can have a number of attributes that determine whether the button is shown, whether the button has a focus, whether text is associated with the

button, and/or whether a number of events are associated with the button, among other attributes. The creation of a design for a GUI can also include defining a number of activities that can be associated with the GUI user controls. For example, a number of activities associated with a button user control can include performing an operation, e.g., addition, subtraction, multiplication, that is associated with a calculator application.

[0011] In a number of previous approaches, the testing stage of the life cycle of an application does not begin until the development stage has concluded. The testing stage and the development stage are performed in a sequential order. In a number of examples of the present disclosure, the testing stage and the development stage can be performed concurrently. A concurrent performance of the testing stage and the development stage can be enabled by a mock-up GUI for testing that allows a mock-up test script to be created while the application is being developed.

[0012] In the present disclosure, reference is made to the accompanying drawings that form a part hereof, and in which is shown by way of illustration how a number of examples of the disclosure can be practiced. These examples are described in sufficient detail to enable those of ordinary skill in the art to practice the examples of this disclosure, and it is to be understood that other examples can be used and that process, electrical, and/or structural changes can be made without departing from the scope of the present disclosure.

[0013] The figures herein follow a numbering convention in which the first digit corresponds to the drawing figure number and the remaining digits identify an element or component in the drawing. Elements shown in the various figures herein can be added, exchanged, and/or eliminated so as to provide a number of additional examples of the present disclosure. In addition, the proportion and the relative scale of the elements provided in the figures are intended to illustrate the examples of the present disclosure, and should not be taken in a limiting sense.

[0014] Figure 1 is a flow chart illustrating an example of a method for test script creation according to the present disclosure. At 102, a functional specification and a UI specification are associated with a proposed application.

[0015] The proposed application can be a plan for how an application will look and how the application will function. The look of the application can be defined by a UI specification. The UI specification can define a number of proposed user controls. The functional specification can define a core function of the proposed application. The core function of a proposed application can be a functional specification that receives input from the UI specification and a functional specification that sends an output to the UI specification.

[0016] As used herein, a proposed application can be independent of a specific programming language. A proposed application defines the UI specification, the proposed user controls, and/or the functional specification generally because the proposed application is not identified with a specific programming language. For example, general details about a user control can define that the user control is to be a button user control and that one of the properties of the button user control is a string with the value "login." However, a proposed application may not define the position of the button user control within the UI specification and/or a number of other attributes that can be associated with the button user control.

[0017] At 104, a mock-up GUI for testing that includes a mock-up GUI based on the UI specification is created. A mock-up GUI for testing may be developed without associating the functional specification with the mock-up GUI. For example, the mock-up GUI can receive input from a user and not forward the input to an associated logic unit. A mock-up GUI can be a template that mirrors an applications GUI. A mock-up GUI can include a number of abstract test user controls. An abstract test user control that is associated with a mock-up GUI can be different than a user control that is associated with a GUI. Figure 3 shows an example of a mock-up GUI for testing.

[0018] A user control can be associated with a particular class in an object oriented language or a collection of attributes. A user control can be a button user control, a radial button user control, and/or an image user control among other types of user controls. For example, a button user control can be a user control with a button type. A type can be associated with a specific class and/or method that defines the attributes that are associated with the user

control. For example, a button user control can have a label attribute and/or an event attribute, among other attributes. A label attribute describes text that is displayed on a button user control. An event attribute describes an event that is initiated when a user selects, e.g., user clicks on, the button user control. In contrast, a text field user control can have a text attribute.

[0019] The method that is associated with a user control and/or a type can also define the naming convention of the attributes and/or the user control as specified by a programming language. For example, a button user control in a first programming language can follow a first naming convention. For example, a first naming convention can be "Button (String label)", wherein "Button" is the naming convention of the user control and "String label" is the naming convention of the label attribute, among other attributes. A button user control in a second programming language can follow a second naming convention. For instance, a second naming convention can be "<Control type='button'>ClickMe</Control>", wherein "<Control type='button'></Control>" is the naming convention of the user control and "ClickMe" is the text that is displayed on the user control.

[0020] In contrast, an abstract test user control can be a representation of the user control that are of a specific type, e.g., button, label, checkbox, among other types, wherein the abstract test user control contains the generalized attributes. In a number of examples, the abstract test user control can be divided into different abstract test user control, e.g., button abstract test user control and label abstract test user control, among other types of abstract test user controls. For example, an abstract test user control can have a button type, a label type, and/or a text field type. However, the abstract test user control with a button type can have a limited set of attributes. For example, a button user control that is an abstract test user control can have only one attribute, a label attribute, and no other attributes. The limited set of attributes associated with an abstract test user control can define general attributes that are relevant to all user controls of a specific type in a number of different programming languages. For example, a limited set of attributes associated with an abstract test user

control with a button type can define general attributes that are relevant to all user controls of button type in a number of different programming languages.

[0021] In a number of examples, an abstract test user control can be associated with a single type and all abstract test user controls can have a limited set of attributes. For example, all abstract user controls can have a text attribute and/or a show attribute or a sting attribute and/or an isEnabled attribute, a label attribute, and/or an event attribute, for example. A label attribute can be text that is associated with the abstract user control type and an isEnabled attribute can determine whether the button can be clicked, for instance. An event attribute can define an action that is taken when a user interacts with an associated abstract test user control. Other limited sets of attributes can be associated with an abstract test user control.

[0022] At 106, an application based on the proposed application that includes a logic based on the functional specification and a GUI based on the UI specification can be implemented. In a number of examples, a mock-up GUI for testing can be developed is less time than an application can be developed because a mock-up GUI for testing does not include a representation of the functional specification and because the mock-up GUI for testing includes a limited representation of the UI specification. For example, a proposed login application that authenticates a user can have a functional specification that is associated with determining whether the user has a number of privileges. The mock-up GUI for testing can include a mock-up GUI that has a number of abstract test user controls that are associated with receiving information from a user and not the functional specification that determines whether the information provided from the user coincides with information that is associated with a user that has a number of privileges. Furthermore, a mock-up GUI can be based on a UI specification without including a representation of all of the proposed user controls that are associated with the UI specification whereas an application can include a representation of all of the proposed user control in the form of user controls.

[0023] At 108, a mock-up test script based on the mock-up GUI and not based on the logic, the functional specification, and the GUI, can be created.

The development of the mock-up test script at 108 can begin while the application is being developed at 106. In a number of examples, development of the mock-up test script can conclude before development of the application concludes. In various examples, development of the mock-up test script can conclude after development of the application concludes.

[0024] A mock-up test script and a test script can be created manually and/or interactively. For example, a user can create a mock-up test script and a test script by developing machine-readable instructions (MRI) that can be used to interact with a mock-up GUI and/or the GUI that is associated with the application. A mock-up test script can also be created by recording a user's interactions with the mock-up GUI for testing. A test script can be created by recording a user's interactions with the application. Interacting with the mock-up GUI can provide a structure that can be used to test the application. For example, interacting with the mock-up GUI can be associated with a mock-up test script that includes a structure with a first step of entering text into a first abstract test user control, a second step of entering text into a second abstract test user control, and/or a third step of selecting a third abstract test user control.

[0025] A test script and/or a mock-up test script can define a number of interactions with the abstract architecture and/or the application. A test script and/or a mock-up test script can also define a number of results that are associated with the abstract architecture and/or the application. For example, a structure that is associated with the mock-up test script can have expected results if applied to a mock-up GUI that includes expected results and/or a test script can have a particular expected result if applied to a GUI that is associated with the application.

[0026] In a number of examples, the mock-up test script can be executed on the mock-up GUI for testing while the test script can be executed on the application. The test script can be executed on the application by converting a number of abstract test user controls, in the mock-up test script, that are associated with the mock-up GUI for testing to user controls that are associated with a specific programming language.

[0027] The conversion can include associating abstract test user controls with corresponding user controls. The association can be based on the attributes associated with the abstract test user control and the attributes associated with the user control. For example, an abstract test user control that has a text attribute with a "Name" value can be associated with a user control that has a label attribute with a "Name" value because a correlation can be made between the text attribute and the label attribute.

[0028] The association of a number of the attributes, e.g., text attribute, can be enhanced with a user's input and/or can be performed without a user's input. For example, a user can indicate which abstract test user controls are associated with which user controls or the comparison can be performed without a user's input. In a number of examples, a predefined mapping, e.g., attribute mapping, can map a number of attributes of the abstract test user control to a number of attributes of the user controls as defined by a number of support programming languages. The association can search the attribute map and associate an attribute from the abstract user control with a corresponding attribute for a user control.

[0029] The conversion can include replacing an abstract test user control with a user control such that the abstract test user control is deleted and the user control is used in its place. The conversion can also include redefining the abstract test user control as a user control such that the abstract test user control is not deleted but changed to reflect the user control.

[0030] Figure 2 is a flow chart illustrating test script creation according to the present disclosure. Figure 1 includes the development of a proposed application 270, a development of a functional architecture 274, a development of a mock-up GUI for testing 272, a development of a mock-up test script 276, and a development of a test script 278.

[0031] In Figure 2 a time associated with the development of the application 270, a development of the mock-up GUI for testing 272, the development of an application 274, the development of a mock-up test script 276, and the development of a test script 278 is indicated by the length of the rectangle. For example, the development of a mock-up GUI for testing 272 can

take less time than the time associated with the development of an application 274.

[0032] The development of a mock-up GUI for testing 272 and the development of a mock-up test script 276 can be performed concurrently with the development of an application 274. Developing the mock-up GUI for testing 272 and the mock-up test script 276 concurrently with the development of the application 274 can reduce the amount of time associated with conducting a number of tests on the application. A mock-up test script can be created based on the mock-up GUI for testing. A development of a test script 278 can begin after the development of the mock-up test script 276 and the development of the application 274 is concluded.

[0033] Figure 3 is a diagram illustrating an example of a mock-up GUI for testing according to the present disclosure. A mock-up GUI for testing can include a mock-up GUI 300. A mock-up GUI 300 includes a number of abstract test user controls 312-1, 312-2, 312-3, 312-4, and 312-5, e.g., referred to generally as abstract test user controls 312.

[0034] The mock-up GUI 300 can be based on a proposed application and/or on a UI specification. The abstract test user controls 312 can be based on a number of proposed user controls. In the example of Figure 3, the proposed application can be a login application.

[0035] The proposed user controls can include a number of text fields, labels, buttons, and/or graphics. The abstract test user controls 312 include abstract test user controls 312-1 and 312-3 that can represent text fields. The abstract test user control 312-1 can have a text attribute with a “Name” value and a show attribute with a “false” value, e.g., indicates that the text attribute is not displayed on the abstract test user control 312-1. The abstract test user control 312-3 can have a text attribute with a “Password” value and a show attribute with a “false” value.

[0036] The abstract test user controls 312-2 and 312-4 can represent labels. The abstract test user control 312-2 can have a text attribute with a “Name” value and a show attribute with a “true” value, e.g., indicates that the text attribute is displayed on the abstract test user control 312-2. The abstract

test user control 312-4 can have a text attribute with a "Password" value and a show attribute with a "true" value. The abstract test user control 312-5 can represent a button. The abstract test user control 312-5 can have a text attribute with a "Login" value and a show attribute with a "true" value.

[0037] The mock-up GUI 300 can receive a number of user interactions. For example, the mock-up GUI 300 can allow a user to select, e.g., click, the abstract test user control 312-5. However, mock-up GUI 300 can be limited in that selecting the abstract test user control 312-5 can have no associated event, e.g., that is no representation of the functional specification.

[0038] The mock-up GUI 300 can include a representation, in the form of abstract test user controls 312, of some of the proposed user controls but not all of the proposed user controls. For example, the proposed application can define a cancel button, and graphics that are not represented in the mock-up GUI 300. Furthermore, the proposed application can define a login button that includes an image instead of the value "Login" as depicted in abstract test user controls 312-5 in mock-up GUI 300.

[0039] A mock-up test script can be created based on the mock-up GUI. For example, a user can interact with the mock-up GUI. A user's interaction can be recorded and associated with the mock-up test script. A user, for instance, can perform a first selection, e.g., a click, on the abstract test user control 312-1, a second selection on the abstract test user control 312-3, and a third selection on the abstract test user control 312-5. The first selection can be performed before the second selection which is performed before the third selection. The first selection, the second selection, and the third selection can be a structure, e.g., sequence of events, that is associated with the mock-up test script and/or the abstract test user controls 312.

[0040] In a number of examples, the mock-up test script can define a number of events initiated by a user and may check the expected results from the user defined events. For example, the structure can define the events initiated by a user and check the expected results of those events if the mock-up GUI 300 contains the result of the action. In a number of examples, the test

script can define a number of events initiated by a user and a number of expected results.

[0041] Figure 4 is a diagram illustrating an example of an application according to the present disclosure. An application can be based on the proposed application. An application can include logic and a GUI 401. The GUI 401 is different than the mock-up GUI 300 in Figure 3 because representations of all of the proposed user controls that are described in the proposed application are included in the GUI 401 in the form of user controls. The user controls include a window user control 420, an image user control 424, text field user control 426, a password field user control 428, label user controls 430-1 and 430-2, and button user controls 432-1 and 432-2.

[0042] Each of the user controls can be associated with a class and/or method that are particular to a programming language. The class and/or method associated with the user controls can define a specific set of attributes. The label user controls 430-1 and 430-2 can have an attribute that defines a number of characters in the text that is associated with the label. The label user control 430-1 can have a text attribute with a "User Name" while the label user control 430-2 can have a text attribute with a "Password" value. The text field user control 426 can have a text attribute that can initially be empty, e.g. a "" value. The password field user control 428 can have a text attribute that can also initially be empty. The button user control 432-1 can have an image attribute that has an image with an arrow as shown. The button user control 432-2 can have an image attribute that has an image with an "x" as shown. The image user control 424 can have an image attribute with a user icon. The window user control 420 can be a rectangle with a length attribute and a width attribute. Each of the user controls can have more attributes than those listed. For example, each of the user controls can have a location attribute that defines the location of the user control within the GUI 401.

[0043] A mock-up test script that is based on the mock-up GIU for testing in Figure 3 can be associated with a number of abstract test user controls, wherein the abstract test user controls are converted to user controls to create the test script. For example, the abstract test user control 312-1 in Figure 3 can

be converted to the text field user control 426, the abstract test user control 312-3 in Figure 3 can be converted to the password field user control 428, the abstract test user control 312-2 in Figure 3 can be converted to the label user control 430-1, the abstract test user control 312-4 in Figure 3 can be converted to the label user control 430-2, and the abstract test user control 312-5 in Figure 3 can be converted to a button user control 432-1. Each of the abstract test user controls 312 in Figure 3 can be converted to user controls based on the attributes that are associated with the abstract test user controls 312 in Figure 3 and the user controls. For example, the abstract test user control 312-2 that has a label attribute with a "Name" value can be associated with the label user control 430-1 that has a text attribute with a "User Name" value because the label attributes and the text attributes can be correlated based on their values. Furthermore, in cases in which an abstract test user control cannot be correlated with a user control based on their attribute, a user can provide input to correlate specific abstract test user controls with user controls.

[0044] The mock-up test script can further be enhanced before and/or after the abstract test user controls are converted. For example, a structure that is associated with the mock-up test script can be modified to include user controls that are not represented in the mock-up test script in the form of abstract test user controls to create the test script. The structure can be modified by including a selection of the button 432-2. The structure can also be modified by adding detail to the structure such as text input into the text field 426 and text input into the password field 428. The mock-up test script can be also be modified by including a number of expected results that are associated with the structure. The test script can then be used to test the application.

[0045] Figure 5 is a diagram illustrating an example of a computing system according to the present disclosure. The computing system 556 can utilize software, hardware, firmware, and/or logic to perform a number of functions.

[0046] The computing system 556 can be a combination of hardware and program instructions configured to perform a number of functions, e.g., actions. The hardware, for example, can include one or more processing resources 554

and other memory resources 544, etc. The program instructions, e.g., machine-readable instructions (MRI), can include instructions stored on memory resource 544 to implement a particular function, e.g., an action such as test script creation.

[0047] The processing resources 554 can be in communication with the memory resource 544 storing the set of MRI executable by one or more of the processing resources 554, as described herein. The MRI can also be stored in a remote memory managed by a server and represent an installation package that can be downloaded, installed and executed. A computing device 556, e.g., server, can include memory resources 544, and the processing resources 554 can be coupled to the memory resources 544 remotely in a cloud computing environment.

[0048] Processing resources 554 can execute MRI that can be stored on internal or external non-transitory memory 544. The processing resources 554 can execute MRI to perform various functions, e.g., acts, including the functions described herein among others.

[0049] As shown in Figure 5, the MRI can be segmented into a number of modules, e.g., a proposed application module 546, a mock-up GUI for testing module 548, an application module 550, and a test script module 552, that when executed by the processing resource 554 can perform a number of functions. As used herein a module includes a set of instructions included to perform a particular task or action. The number of modules 546, 548, 550, and 552 can be sub-modules of other modules. For example, the proposed application module 546 and the mock-up GUI for testing module 548 can be sub-modules and/or contained within a single module. Furthermore, the number of modules 546, 548, 550, and 552 can comprise individual modules separate and distinct from one another.

[0050] In the example of Figure 5, a proposed application module 546 can comprise MRI that are executed by the processing resources 554 to define a proposed application. The definition of a proposed application can include the definition of a UI specification and proposed user controls that are associated with the GUI. Defining proposed user controls can include defining a number of

proposed attributes that are associated with the user control and/or events that are associated with the proposed user controls. Defining a proposed application can also include defining a functional specification that defines a number of core functions of the application.

[0051] A mock-up GUI for testing 548 can comprise MRI that are executed by the processing resources 554 to create a mock-up GUI. An abstract GUI for testing can include a mock-up GUI and abstract test user control that are associated with the mock-up GUI. The abstract test user controls have a limited number of attributes as compared to user controls that are associated with a GUI in the application. Abstract test user controls are different than user controls in that user controls are defined specific to a programming language while abstract test user control can be converted into a user control as defined by any number of programming languages. A mock-up GUI for testing 548 can be created concurrently with the creation of the application.

[0052] An application module 550 can comprise MRI that are executed by the processing resources 554 to create an application. An application can include logic that is based on the functional specification and a GUI that is based on the UI specification. An application can be created specific to a programming language. The programming language can dictate a number of attributes that are associated with user controls in the GUI.

[0053] Test script module 552 can comprise MRI that are executed by the processing resources 554 to create a test script. The mock-up test script can be created concurrently with the creation of the application. A mock-up test script can be based on the mock-up GUI for testing. A mock-up test script includes a structure, e.g., events initiated by a user on abstract test user controls in the mock-up GUI, that is associated with a number of abstract test user controls. The abstract test user controls that are associated with the structure and the mock-up test script can be converted to user controls to create a test script. The conversion can be based on the attributes associated with the abstract test user controls and the user controls. Attributes can include events which can be taken on the abstract test user controls and/or user controls, features of the

abstract test user controls, and/or user controls, and/or characteristics of the abstract test user controls and/or user controls, among other types of attributes. After the conversion, the test script can be used to test the application.

[0054] A memory resource 544, as used herein, can include volatile and/or non-volatile memory. Volatile memory can include memory that depends upon power to store information, such as various types of dynamic random access memory (DRAM) among others. Non-volatile memory can include memory that does not depend upon power to store information. Examples of non-volatile memory can include solid state media such as flash memory, electrically erasable programmable read-only memory (EEPROM), phase change random access memory (PCRAM), magnetic memory such as a hard disk, tape drives, floppy disk, and/or tape memory, optical discs, digital versatile discs (DVD), Blu-ray discs (BD), compact discs (CD), and/or a solid state drive (SSD), etc., as well as other types of computer-readable media.

[0055] The memory resource 544 can be integral or communicatively coupled to a computing device in a wired and/or wireless manner. For example, the memory resource 544 can be an internal memory, a portable memory, and a portable disk, or a memory associated with another computing resource, e.g., enabling MRIs to be transferred and/or executed across a network such as the Internet.

[0056] The memory resource 544 can be in communication with the processing resources 554 via a communication path 560. The communication path 560 can be local or remote to a machine, e.g., a computer, associated with the processing resources 554. Examples of a local communication path 560 can include an electronic bus internal to a machine, e.g., a computer, where the memory resource 544 is one of volatile, non-volatile, fixed, and/or removable storage medium in communication with the processing resources 554 via the electronic bus. Examples of such electronic buses can include Industry Standard Architecture (ISA), Peripheral Component Interconnect (PCI), Advanced Technology Attachment (ATA), Small Computer System Interface (SCSI), Universal Serial Bus (USB), among other types of electronic buses and variants thereof.

[0057] The communication path 560 can be such that the memory resource 544 is remote from a processing resource, e.g., processing resources 554, such as in a network connection between the memory resource 544 and the processing resource, e.g., processing resources 554. That is, the communication path 560 can be a network connection. Examples of such a network connection can include local area network (LAN), wide area network (WAN), personal area network (PAN), and the Internet, among others. In such examples, the memory resource 544 can be associated with a first computing device and the processing resources 554 can be associated with a second computing device, e.g., a Java® server. For example, processing resources 554 can be in communication with a memory resource 544, wherein the memory resource 544 includes a set of instructions and wherein the processing resources 554 are designed to carry out the set of instructions.

[0058] As used herein, “a” or “a number of” something can refer to one or more such things. For example, “a number of widgets” can refer to one or more widgets.

[0059] The above specification, examples and data provide a description of the method and applications, and use of the system and method of the present disclosure. Since many examples can be made without departing from the spirit and scope of the system and method of the present disclosure, this specification merely sets forth some of the many possible embodiment configurations and implementations.

What is claimed:

1. A system for test script creation, comprising:

a processing resource in communication with a memory resource, wherein the memory resource includes a set of instructions, executable by the processing resource to:

create a mock-up GUI for testing, wherein the mock-up GUI for testing includes a number of abstract test user controls;

implement an application that includes a logic, and a GUI, wherein the application includes a number of user controls;

create the mock-up test script based on the mock-up GUI and the number of abstract test user controls before the application is developed; and

replace the number of abstract test user controls in the mock-up test script with the number of user controls to create a test script;

test the application by implementing the test script on the application and on the number of user controls.

2. The system of claim 1, wherein the instructions executable to replace the number of abstract test user controls in the mock-up test script with the number of user controls include instructions to associate the number of abstract test user controls with the number of user controls.

3. The system of claim 2, wherein the instructions executable to associate the abstract test user controls with the user controls include instruction to associate a first number of properties that are associated with the number of abstract test user controls with a second number of properties that are associated with the number of user controls.

4. The system of claim 3, wherein the instructions are executable to replace the number of abstract test user controls in the mock-up test script with the number of user controls based on the association of the number of abstract test user controls with the number of user controls.

5. The system of claim 1, wherein the instructions are executable to modify the mock-up test script after is it created to create the test script that is based on the logic and the GUI before the application is tested.

6. A non-transitory machine-readable medium storing instructions for test script creation executable by a computer to cause the computer to:

create a functional specification and a user interface (UI) specification for a proposed application;

implement a mock-up GUI for testing based on the UI specification, wherein the mock-up GUI for testing is not based on the functional specification;

develop an application based on the proposed application that includes a logic based on the functional specification and a GUI based on the UI specification;

create a mock-up test script based on the mock-up GUI and not based on the logic, the functional specification, and the GUI; and

test the mock-up GUI by executing the mock-up test script on the mock-up GUI.

7. The medium of claim 6, wherein the instructions executable to implement the mock-up GUI for testing include instructions to create a number of abstract test user controls that are part of the mock-up GUI wherein the number of abstract test user controls are defined by a number of abstract properties.

8. The medium of claim 7, wherein the instructions executable to develop the application include instructions to create a number of user controls that are part of the GUI wherein the number of user controls are defined by a number of properties.

9. The medium of claim 8, wherein the number of abstract test user controls represent the number of user controls that are language specific.

10. The medium of claim 6, wherein the instructions executable to create the mock-up test script include instructions to record a user's manipulation of an input device to perform a number of events on the number of abstract test user controls.

11. A method for test script creation comprising:

creating a functional specification and a user interface (UI) specification for a proposed application;

implementing a mock-up GUI for testing based on the UI specification;

developing an application based on the proposed application that includes a logic based on the functional specification and a GUI based on the UI specification; and

creating a mock-up test script based on the proposed application and the mock-up GUI and not based on the logic, the functional specification, and the GUI.

12. The method of claim 11, wherein creating the mock-up GUI for testing include creating the mock-up GUI for testing while the application is being developed.

13. The method of claim 11, wherein creating the mock-up test script includes creating the mock-up test script while the application is being developed.

14. The method of claim 13, wherein creating the mock-up test script based on the functional application includes creating the mock-up test script after the functional application is created.

15. The method of claim 11, wherein the proposed application is a plan for how the application will look and how the application will function.

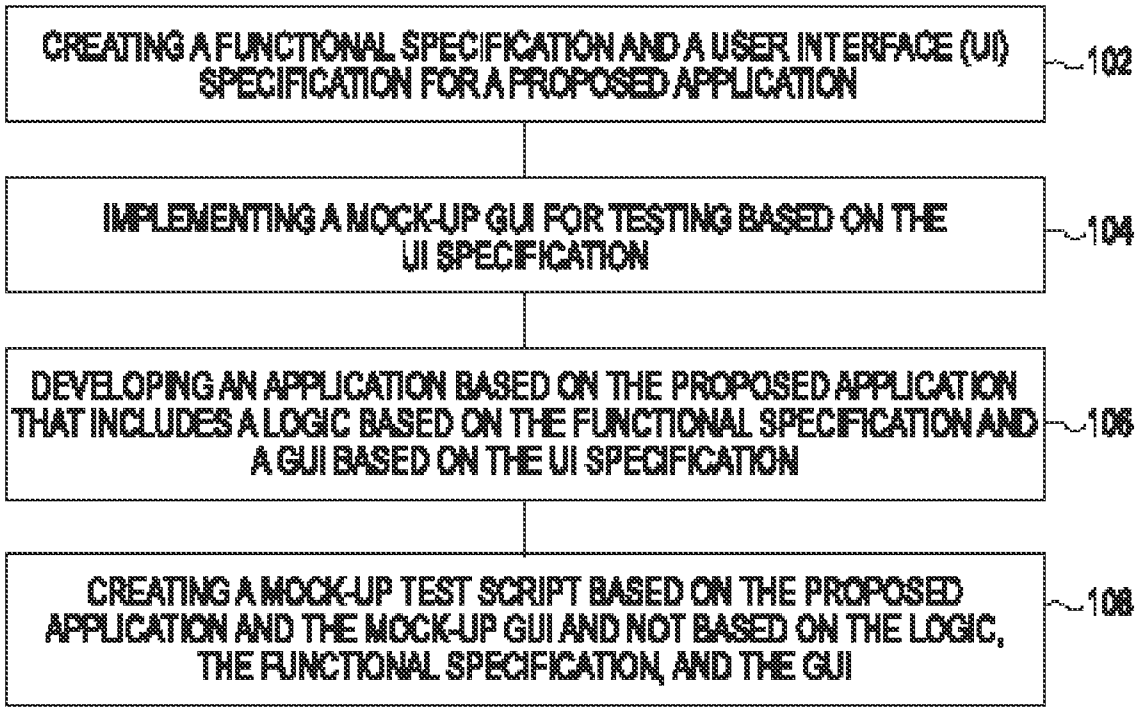
CREATING A FUNCTIONAL SPECIFICATION AND A USER INTERFACE (UI) SPECIFICATION FOR A PROPOSED APPLICATION
102
IMPLEMENTING A MOCK-UP GUI FOR TESTING BASED ON THE UI SPECIFICATION
104
DEVELOPING AN APPLICATION BASED ON THE PROPOSED APPLICATION THAT INCLUDES A LOGIC BASED ON THE FUNCTIONAL SPECIFICATION AND A GUI BASED ON THE UI SPECIFICATION
106
CREATING A MOCK-UP TEST SCRIPT BASED ON THE PROPOSED APPLICATION AND THE MOCK-UP GUI AND NOT BASED ON THE LOGIC, THE FUNCTIONAL SPECIFICATION, AND THE GUI
108

Fig. 1

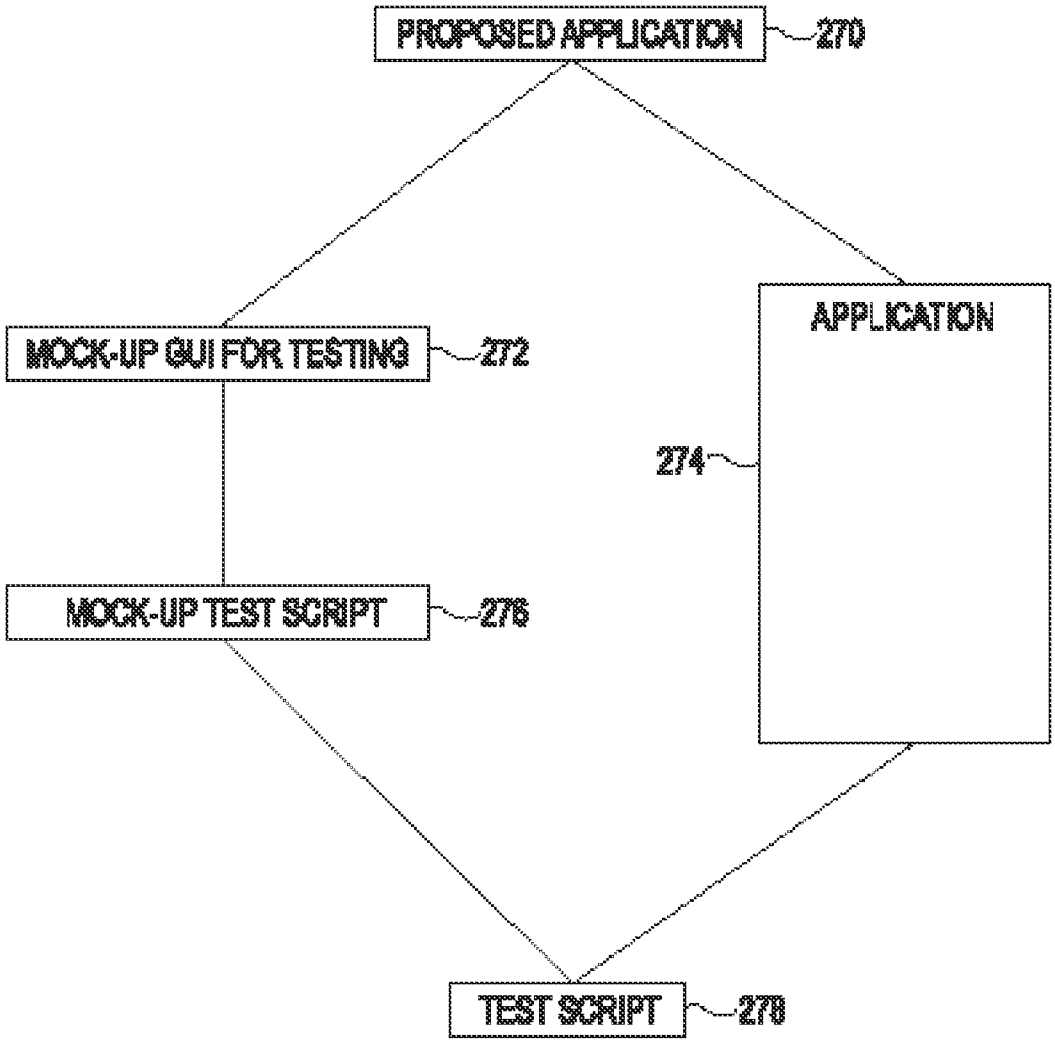
PROPOSED APPLICATION
270
MOCK-UP GUI FOR TESTING
272
APPLICATION
274
MOCK-UP TEST SCRIPT
276
TEST SCRIPT
278

Fig. 2

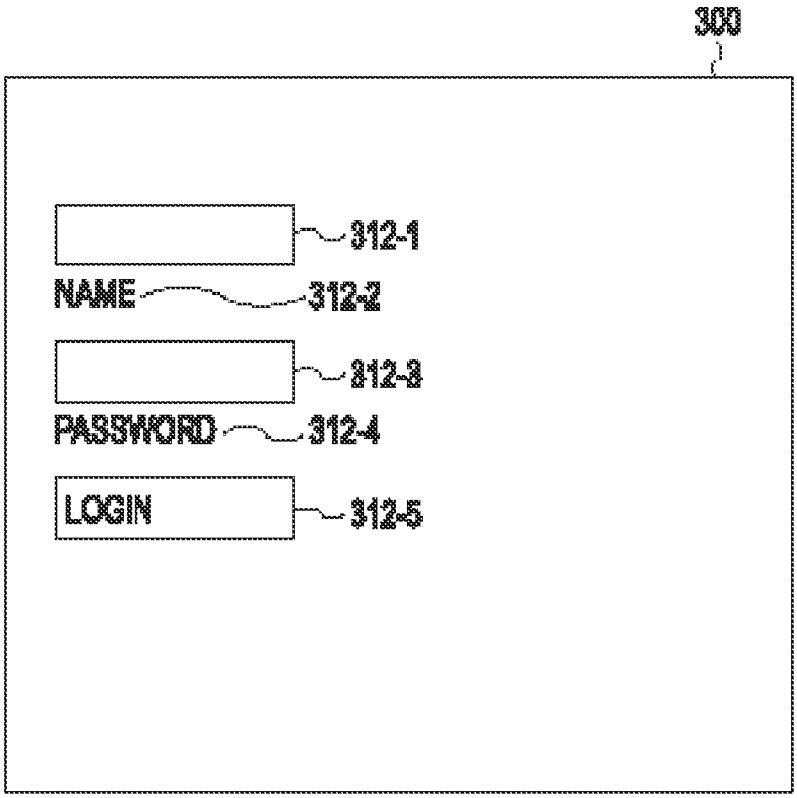

Fig. 3

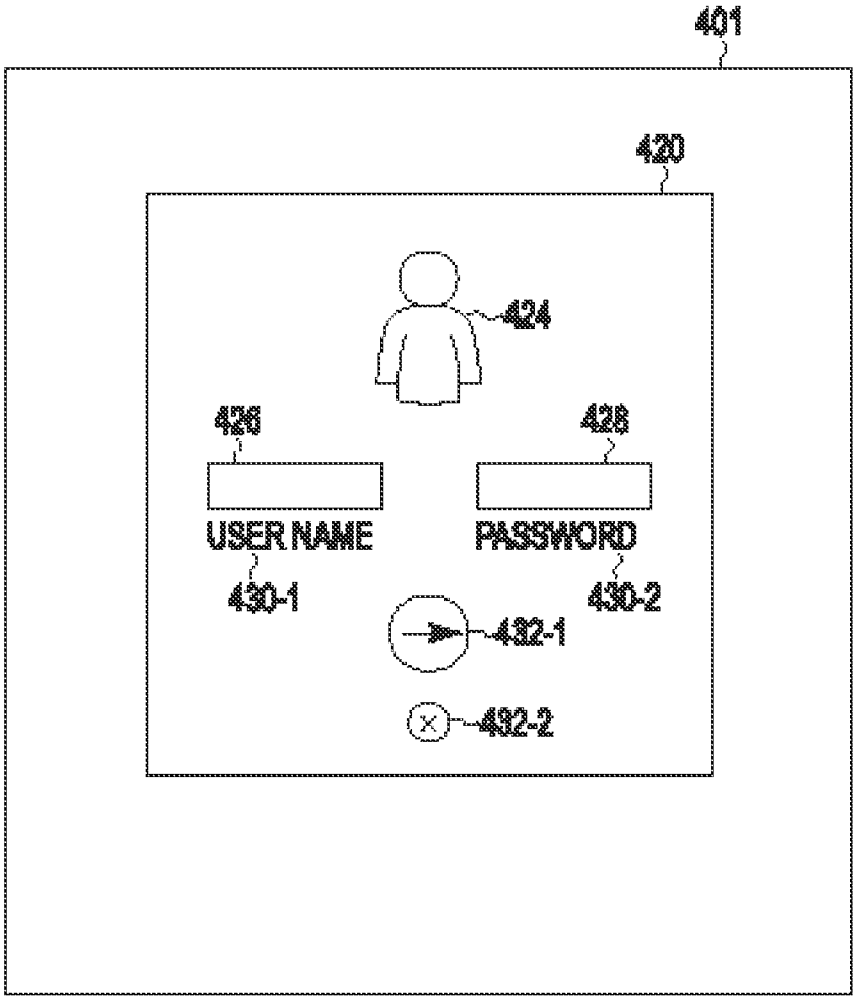

Fig. 4

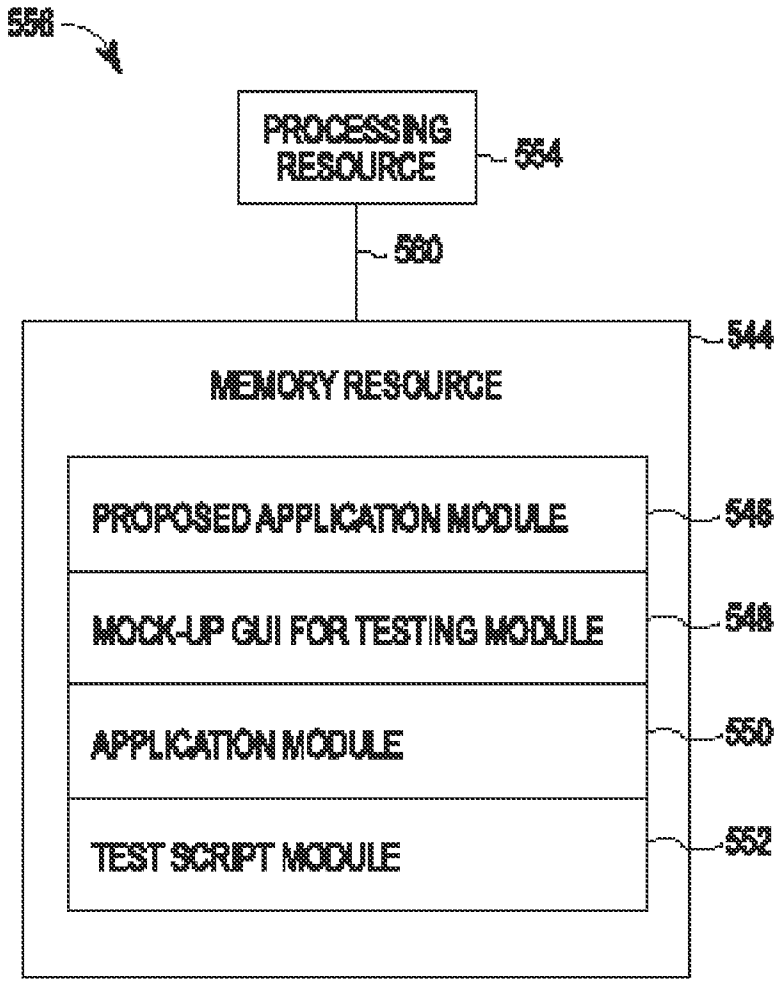
556
PROCESSING RESOURCE
554
560
544
MEMORY RESOURCE
PROPOSED APPLICATION MODULE
546
MOCK-UP GUI FOR TESTING MODULE
548
APPLICATION MODULE
550
TEST SCRIPT MODULE
552

Fig. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2013/071264

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/46 (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)

IPC: G06F 9/-; G06F 11/36

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)

VEN, CNABS, CNTXT, ENTXT, CNKI: MOCK+, MODEL+, UI, USER INTERFACE, HMI, HUMAN MACHINE INTERFACE, TEST+, DEBUG+, SCRIPT, CREAT+, SOFTWARE, APPLICATION, ABSTRACT, REPLAC+, CONVERT+, MAP+, TRANSFORM+

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 101425012 A (SHENZHEN HUAWEI COMMUNICATION TECHNOLOGY) 06 May 2009 (06.05.2009) the whole document	1-15
A	US 2010037210 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 11 February 2010 (11.02.2010) the whole document	1-15
A	US 6986125 B2 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 10 January 2006 (10.01.2006) the whole document	1-15

☐ 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 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

"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	Date of mailing of the international search report
06 October 2013(06.10.2013)	**07 Nov. 2013 (07.11.2013)**
Name and mailing address of the ISA/CN The State Intellectual Property Office, the P.R.China 6 Xitucheng Rd., Jimen Bridge, Haidian District, Beijing, China 100088 Facsimile No. 86-10-62019451	Authorized officer WU, Shifen Telephone No. (86-10)61041035

Form PCT/ISA /210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2013/071264

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 101425012 A	06.05.2009	None	
US 2010037210 A1	11.02.2010	JP 4148527 B2	10.09.2008
		JP 2007323573 A	13.12.2007
		US 2008072100 A1	20.03.2008
		US 7624380 B2	24.11.2009
		US 7996819 B2	09.08.2011
US 6986125 B2	10.01.2006	US 2003037314 A1	20.02.2003