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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 July 2014 (03.07.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/101585 A1

(51) International Patent Classification:
G06F 9/44 (2006.01)

(21) International Application Number:
PCT/CN2013/087380

(22) International Filing Date:
19 November 2013 (19.11.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
201210567720.7 24 December 2012 (24.12.2012) CN

(71) Applicant: **TENCENT TECHNOLOGY (SHENZHEN) COMPANY LIMITED** [CN/CN]; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian, Shenzhen, Guangdong 518000 (CN).

(72) Inventor: **LU, Sixi**; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian, Shenzhen, Guangdong 518000 (CN).

(74) Agent: **BEIJING SAN GAO YONG XIN INTELLECTUAL PROPERTY AGENCY CO., LTD**; A-1-102, He Jing Yuan, Ji Men Li, Xueyuan Road, Haidian, Beijing 100088 (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, IR, IS, JP, KE, KG, 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, SA, 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, KM, ML, MR, NE, SN, TD, TG).

Published:

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

(54) **Title**: SYSTEMS AND METHODS FOR GENERATING FUNCTION-RELATION CALL TREES

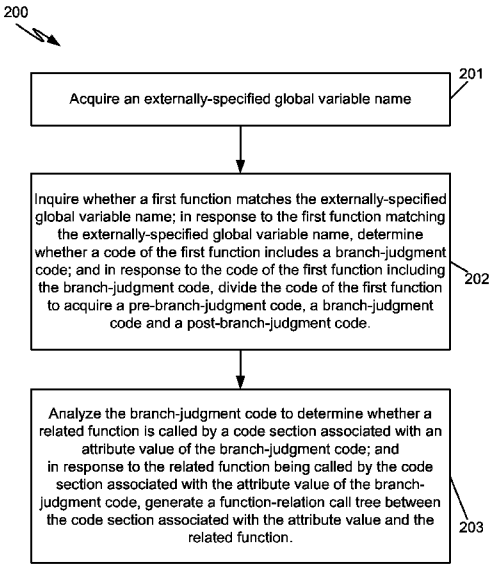

Figure 2

(57) **Abstract**: Systems and methods are provided for generating a function-relation call tree. For example, an externally-specified global variable name is acquired; whether a first function matches the externally-specified global variable name is inquired; in response to the first function matching the externally-specified global variable name, whether a code of the first function includes a branch-judgment code is determined; in response to the code of the first function including the branch-judgment code, the code of the first function is divided to acquire a pre-branch-judgment code, a branch- judgment code and a post-branch-judgment code; the branch-judgment code is analyzed to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code; and in response to the related function being called by the code section associated with the attribute value of the branch-judgment code, a function-relation call tree between the code section associated with the attribute value and the related function is generated.

WO 2014/101585 A1

SYSTEMS AND METHODS FOR GENERATING FUNCTION-RELATION CALL TREES

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] This application claims priority to Chinese Patent Application No. 201210567720.7, filed December 24, 2012, incorporated by reference herein for all purposes.

BACKGROUND OF THE INVENTION

[0002] The present invention is directed to computer technology. More particularly, the invention provides systems and methods for data processing. Merely by way of example, the invention has been applied to computer programs. But it would be recognized that the invention has a much broader range of applicability.

[0003] In software development, when a code of a function is changed, the effects of the code change are often assessed to determine whether the code change may cause incompatibility so as to provide a more accurate test range. In actual application, a function-relation call tree is generally used to assess the effects of the code change. That is, upon the code change, a function associated with the changed code is determined, and then the upper-level call relations and the lower-level call relations are checked for assessing one or more related functions that may be affected by the code change.

[0004] Figure 1 is a conventional simplified diagram showing a function-relation call tree. As shown in Figure 1, when code change is included in a code section "case 'IE'" associated with an attribute value "IE" in Function 1, it is assessed under ideal circumstances that related functions affected by the code change include only "Function2" that has a call relation with the code section "case 'IE'." But the structure of the function-relation call tree as shown in Figure 1 determines that the granularity for assessing the effects of the code change is at the function level. That is, when the code section "case 'IE'" associated with the attribute value "IE" in Function 1 changes, it is assessed that "Function 2" and "Function 3" that have calling relations with "Function 1" are both affected.

[0005] The structure of the function-relation call tree determines that the granularity used for assessing the effects of the code change is very large, which often makes it difficult to accurately determine the effects of the code change associated with a certain attribute value in a function, and increases the burden on the system for assessing the effects of the code change.

[0006] Hence it is highly desirable to improve the techniques for assessing effects of code changes.

BRIEF SUMMARY OF THE INVENTION

[0007] The present invention is directed to computer technology. More particularly, the invention provides systems and methods for data processing. Merely by way of example, the invention has been applied to computer programs. But it would be recognized that the invention has a much broader range of applicability.

[0008] According to one embodiment, a method is provided for generating a function-relation call tree. For example, an externally-specified global variable name is acquired; whether a first function matches the externally-specified global variable name is inquired; in response to the first function matching the externally-specified global variable name, whether a code of the first function includes a branch-judgment code is determined; in response to the code of the first function including the branch-judgment code, the code of the first function is divided to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code; the branch-judgment code is analyzed to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code; and in response to the related function being called by the code section associated with the attribute value of the branch-judgment code, a function-relation call tree between the code section associated with the attribute value and the related function is generated.

[0009] According to another embodiment, a system for generating a function-relation call tree includes an acquisition unit, an inquiry unit, a judgment unit, a splitting unit, a determination unit, and a generation unit. The acquisition unit is configured to acquire an externally-specified global variable name. The inquiry unit is configured to inquire whether a first function matches the externally-specified global variable name. The judgment unit is configured to determine whether a code

of the first function includes a branch-judgment code in response to the first function matching the externally-specified global variable name. The splitting unit is configured to divide the code of the first function to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code in response to the code of the first function including the branch-judgment code. The determination unit is configured to analyze the branch-judgment code to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code. The generation unit is configured to generate a function-relation call tree between the code section associated with the attribute value and the related function in response to the related function being called by the code section associated with the attribute value of the branch-judgment code.

[0010] According to yet another embodiment, a non-transitory computer readable storage medium comprises programming instructions for generating a function-relation call tree. The programming instructions are configured to cause one or more data processors to execute certain operations. For example, an externally-specified global variable name is acquired; whether a first function matches the externally-specified global variable name is inquired; in response to the first function matching the externally-specified global variable name, whether a code of the first function includes a branch-judgment code is determined; in response to the code of the first function including the branch-judgment code, the code of the first function is divided to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code; the branch-judgment code is analyzed to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code; and in response to the related function being called by the code section associated with the attribute value of the branch-judgment code, a function-relation call tree between the code section associated with the attribute value and the related function is generated.

[0011] For example, the systems and methods disclosed herein are configured to generate a function-relation call tree that has a relatively small granularity for assessing effects of code changes so as to accurately determine the effects of the code changes, and reducing the system burden for assessment of the effects of the code changes.

[0012] Depending upon embodiment, one or more benefits may be achieved. These benefits and various additional objects, features and advantages of the present invention can be fully appreciated with reference to the detailed description and accompanying drawings that follow.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Figure 1 is a conventional simplified diagram showing a function-relation call tree;

[0014] Figure 2 is a simplified diagram showing a method for generating a function-relation call tree according to one embodiment of the present invention;

[0015] Figure 3 is a simplified diagram showing a method for generating a function-relation call tree according to another embodiment of the present invention;

[0016] Figure 4 is a simplified diagram showing multiple functions according to another embodiment of the present invention;

[0017] Figure 5 is a simplified diagram showing a system for generating a function-relation call tree according to one embodiment of the present invention; and

[0018] Figure 6 is a simplified diagram showing a system for generating a function-relation call tree according to another embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0019] The present invention is directed to computer technology. More particularly, the invention provides systems and methods for data processing. Merely by way of example, the invention has been applied to computer programs. But it would be recognized that the invention has a much broader range of applicability.

[0020] Figure 2 is a simplified diagram showing a method for generating a function-relation call tree according to one embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the claims. One of ordinary skill in the art would recognize many variations, alternatives, and modifications. The method 200 includes at least the process 201 for acquiring an externally-specified global variable name, the process 202 for inquiring whether a

first function matches the externally-specified global variable name, determining whether a code of the first function includes a branch-judgment code, and dividing the code of the first function to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code, and the process 203 for analyzing the branch-judgment code to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code and generating a function-relation call tree between the code section associated with the attribute value and the related function.

[0021] According to one embodiment, during the process 201, an externally-specified global variable name is acquired. For example, the externally-specified global variable name is received (e.g., by a system). In another example, the externally-specified global variable name is actively read (e.g., by the system).

[0022] According to another embodiment, during the process 202, whether a first function matches the externally-specified global variable name is inquired. For example, in response to the first function matching the externally-specified global variable name, whether a code of the first function includes a branch-judgment code is determined. As an example, in response to the code of the first function including the branch-judgment code, the code of the first function is divided to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code. In an example, the inquiring whether a first function matches the externally-specified global variable name includes: performing a static analysis on the code of the first function to acquire a predefined global variable name associated with the first function and determining whether the predefined global variable name associated with the first function is the same as the externally-specified global variable name. For example, in response to the predefined global variable name associated with the first function being the same as the externally-specified global variable name, the first function is determined to match the externally-specified global variable name. In another example, in response to the predefined global variable name associated with the first function being different from the externally-specified global variable name, the first function is determined not to match the externally-specified global variable name.

[0023] According to yet another embodiment, during the process 203, the branch-judgment code is analyzed to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code. For example, in response to the related function being called by the code section associated with the attribute value of the branch-judgment code, a function-relation call tree between the code section associated with the attribute value and the related function is generated.

[0024] In one embodiment, if the first function is determined to match the externally-specified global variable name during the process 202, and the code of the first function does not include the branch-judgment code, a global call tree for the entire first function is generated. For example, if it is determined during the process 203 that the related function is not called by the code section associated with the attribute value in the branch-judgment code, the global call tree for the entire first function is generated.

[0025] In some embodiments, the method 200 further includes the process for generating a global function-relation call tree for the entire first function based on at least information associated with merging the function-relation call tree between the code section associated with the attribute value and the related function. In certain embodiments, the method 200 further includes the process for predefining one or more global variable names and a set of attribute values and reading the predefined one or more global variable names and one or more attribute values from the set of attribute values into the first function. As an example, the process for predefining one or more global variable names and a set of attribute values and reading the predefined one or more global variable names and one or more attribute values from the set of attribute values into the first function is executed before the process for acquiring the externally-specified global variable name.

[0026] Figure 3 is a simplified diagram showing a method for generating a function-relation call tree according to another embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the claims. One of ordinary skill in the art would recognize many variations, alternatives, and modifications. The method 300 includes at least the process 301 for predefining one or more global variable names and a set of attribute values, the process 302 for reading the predefined one or more global variable names and one or more attribute

values from the set of attribute values into a first function, the process 303 for receiving an externally-specified global variable name, performing a static analysis on a code of the first function, inquiring whether the first function matches the externally-specified global variable name, determining whether the code of the first function includes a branch-judgment code, and dividing the code of the first function to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code, and the process 304 for analyzing the branch-judgment code to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code and generating a function-relation call tree between the code section associated with the attribute value and the related function.

[0027] According to one embodiment, during the process 301, one or more global variable names and a set of attribute values are predefined. For example, the predefined global variable names and the predefined set of attribute values are used in the first function. As an example, a predefined global variable name is "string gBrowser." In another example, a predefined set of attribute values is {"IE", "Chrome", "Firefox"}. The predefined global variable names are placed in a file associated with the first function, for example, a header file of C/C++, in some embodiments.

[0028] According to another embodiment, during the process 302, the predefined one or more global variable names and one or more attribute values are read from the set of attribute values into a first function. For example, a morphology and grammar analyzer is used to read the predefined global variable names and one or more attribute values in the predefined set of attribute values into the first function.

[0029] According to yet another embodiment, during the process 303, an externally-specified global variable name is received, and a static analysis is performed on a code of the first function. For example, whether the first function matches the externally-specified global variable name is inquired. In another example, in response to the first function matching the externally-specified global variable name, whether the code of the first function includes a branch-judgment code is determined. As an example, in response to the code of the first function including the

branch-judgment code, the code of the first function is divided to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code.

[0030] In one embodiment, during the process 304, the branch-judgment code is analyzed to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code. For example, in response to the related function being called by the code section associated with the attribute value of the branch-judgment code, a function-relation call tree between the code section associated with the attribute value and the related function is generated.

[0031] Figure 4 is a simplified diagram showing multiple functions according to another embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the claims. One of ordinary skill in the art would recognize many variations, alternatives, and modifications.

[0032] As shown in Figure 4, it is determined (e.g., during the process 303 as shown in Figure 3) that a function "Function1" matches an externally-specified global variable name and the code of "Function1" includes a branch-judgment code, in some embodiments. For example, the code of "Function1" are divided into a pre-branch-judgment code ("code section 1"), a branch-judgment code ("code section 2") and a post-branch-judgment code ("code section 3"). As an example, further analyses are performed on the branch-judgment code ("code section 2"), and it is determined that two related functions "Function2" and "Function3" are called by code sections with different attribute values "IE" and "Firefox" in the branch-judgment code ("code section 2") respectively. In another example, a function-relation call tree between the code section associated with the attribute value "IE" and the related "Function2" is generated, and a function-relation call tree between the code section associated with the attribute value "Firefox" and the related "Function3" is generated.

[0033] In certain embodiments, if changes are made to the code section associated with the attribute value "IE," the effects of the code change are assessed to be on "Function2" and only the branch from "Function1" to "Function2" needs to be verified. On the other hand, if changes are made to the code section associated with the attribute value "Firefox," the effects of the code change are assessed to be on "Function3" and only the branch from "Function1" to "Function3" needs to be verified. In some embodiments, if changes are made to both the code section

associated with the attribute value "IE" and the code section associated with the attribute value "Firefox," the effects of the code change are assessed to be on "Function2" and "Function3." For example, the branch from "Function1" to "Function2" and the branch from "Function1" to "Function3" need to be verified. As an example, a function-relation call tree is generated using flex, bison and/or graphviz.

[0034] Figure 5 is a simplified diagram showing a system for generating a function-relation call tree according to one embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the claims. One of ordinary skill in the art would recognize many variations, alternatives, and modifications. The system 500 includes an acquisition unit 501, an inquiry unit 502, a judgment unit 503, a splitting unit 504, a determination unit 505, and a generation unit 506.

[0035] According to one embodiment, the acquisition unit 501 is configured to acquire an externally-specified global variable name. For example, the inquiry unit 502 is configured to inquire whether a first function matches the externally-specified global variable name. As an example, the judgment unit 503 is configured to determine whether a code of the first function includes a branch-judgment code in response to the first function matching the externally-specified global variable name. In another example, the splitting unit 504 is configured to divide the code of the first function to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code in response to the code of the first function including the branch-judgment code. In yet another example, the determination unit 505 is configured to analyze the branch-judgment code to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code. For example, the generation unit 506 is configured to generate a function-relation call tree between the code section associated with the attribute value and the related function in response to the related function being called by the code section associated with the attribute value of the branch-judgment code.

[0036] Figure 6 is a simplified diagram showing the system 500 for generating a function-relation call tree according to another embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the claims. One of ordinary skill in the art would recognize many variations, alternatives,

and modifications. The system 500 further includes a merging unit 507, a predefined unit 508, and a reading unit 509. For example, the inquiry unit 502 includes an analysis sub-unit 5021 and a judgment sub-unit 5022.

[0037] According to one embodiment, the analysis sub-unit 5021 is configured to perform a static analysis on the code of the first function to acquire a predefined global variable name associated with the first function. For example, the judgment sub-unit 5022 is configured to determine whether the predefined global variable name associated with the first function is the same as the externally-specified global variable name. In another example, the judgment sub-unit 5022 is further configured to: in response to the predefined global variable name associated with the first function being the same as the externally-specified global variable name, determine that the first function matches the externally-specified global variable name, and in response to the predefined global variable name associated with the first function being different from the externally-specified global variable name, determine that the first function does not match the externally-specified global variable name.

[0038] According to another embodiment, the merging unit is configured to generate a global function-relation call tree based on at least information associated with merging the function-relation call tree between the code section associated with the attribute value and the related function. For example, the predefined unit 508 is configured to predefine one or more global variable names and a set of attribute values. As an example, the reading unit 509 is configured to read the predefined one or more global variable names and one or more attribute values from the set of attribute values into the first function.

[0039] According to yet another embodiment, a method is provided for generating a function-relation call tree. For example, an externally-specified global variable name is acquired; whether a first function matches the externally-specified global variable name is inquired; in response to the first function matching the externally-specified global variable name, whether a code of the first function includes a branch-judgment code is determined; in response to the code of the first function including the branch-judgment code, the code of the first function is divided to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code; the branch-judgment code is analyzed to determine whether a related function is called by

a code section associated with an attribute value of the branch-judgment code; and in response to the related function being called by the code section associated with the attribute value of the branch-judgment code, a function-relation call tree between the code section associated with the attribute value and the related function is generated. For example, the method is implemented according to at least Figure 2, and/or Figure 3.

[0040] According to another embodiment, a system for generating a function-relation call tree includes an acquisition unit, an inquiry unit, a judgment unit, a splitting unit, a determination unit, and a generation unit. The acquisition unit is configured to acquire an externally-specified global variable name. The inquiry unit is configured to inquire whether a first function matches the externally-specified global variable name. The judgment unit is configured to determine whether a code of the first function includes a branch-judgment code in response to the first function matching the externally-specified global variable name. The splitting unit is configured to divide the code of the first function to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code in response to the code of the first function including the branch-judgment code. The determination unit is configured to analyze the branch-judgment code to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code. The generation unit is configured to generate a function-relation call tree between the code section associated with the attribute value and the related function in response to the related function being called by the code section associated with the attribute value of the branch-judgment code. For example, the system is implemented according to at least Figure 5, and/or Figure 6.

[0041] According to yet another embodiment, a non-transitory computer readable storage medium comprises programming instructions for generating a function-relation call tree. The programming instructions are configured to cause one or more data processors to execute certain operations. For example, an externally-specified global variable name is acquired; whether a first function matches the externally-specified global variable name is inquired; in response to the first function matching the externally-specified global variable name, whether a code of the first function includes a branch-judgment code is determined; in response to the code of the first function including the branch-judgment code, the code of the first function is divided

to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code; the branch-judgment code is analyzed to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code; and in response to the related function being called by the code section associated with the attribute value of the branch-judgment code, a function-relation call tree between the code section associated with the attribute value and the related function is generated. For example, the storage medium is implemented according to at least Figure 2, and/or Figure 3.

[0042] The above only describes several scenarios presented by this invention, and the description is relatively specific and detailed, yet it cannot therefore be understood as limiting the scope of this invention's patent. It should be noted that ordinary technicians in the field may also, without deviating from the invention's conceptual premises, make a number of variations and modifications, which are all within the scope of this invention. As a result, in terms of protection, the patent claims shall prevail.

[0043] For example, some or all components of various embodiments of the present invention each are, individually and/or in combination with at least another component, implemented using one or more software components, one or more hardware components, and/or one or more combinations of software and hardware components. In another example, some or all components of various embodiments of the present invention each are, individually and/or in combination with at least another component, implemented in one or more circuits, such as one or more analog circuits and/or one or more digital circuits. In yet another example, various embodiments and/or examples of the present invention can be combined.

[0044] Additionally, the methods and systems described herein may be implemented on many different types of processing devices by program code comprising program instructions that are executable by the device processing subsystem. The software program instructions may include source code, object code, machine code, or any other stored data that is operable to cause a processing system to perform the methods and operations described herein. Other implementations may also be used, however, such as firmware or even appropriately designed hardware configured to carry out the methods and systems described herein.

[0045] The systems' and methods' data (e.g., associations, mappings, data input, data output, intermediate data results, final data results, etc.) may be stored and implemented in one or more different types of computer-implemented data stores, such as different types of storage devices and programming constructs (e.g., RAM, ROM, Flash memory, flat files, databases, programming data structures, programming variables, IF-THEN (or similar type) statement constructs, etc.). It is noted that data structures describe formats for use in organizing and storing data in databases, programs, memory, or other computer-readable media for use by a computer program.

[0046] The systems and methods may be provided on many different types of computer-readable media including computer storage mechanisms (e.g., CD-ROM, diskette, RAM, flash memory, computer's hard drive, etc.) that contain instructions (e.g., software) for use in execution by a processor to perform the methods' operations and implement the systems described herein.

[0047] The computer components, software modules, functions, data stores and data structures described herein may be connected directly or indirectly to each other in order to allow the flow of data needed for their operations. It is also noted that a module or processor includes but is not limited to a unit of code that performs a software operation, and can be implemented for example as a subroutine unit of code, or as a software function unit of code, or as an object (as in an object-oriented paradigm), or as an applet, or in a computer script language, or as another type of computer code. The software components and/or functionality may be located on a single computer or distributed across multiple computers depending upon the situation at hand.

[0048] The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

[0049] While this specification contains many specifics, these should not be construed as limitations on the scope or of what may be claimed, but rather as descriptions of features specific to particular embodiments. Certain features that are described in this specification in the context or separate embodiments can also be

implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[0050] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various system components in the embodiments described above should not be understood as requiring such separation in all embodiments, and it should be understood that the described program components and systems can generally be integrated together in a single software product or packaged into multiple software products.

[0051] Although specific embodiments of the present invention have been described, it will be understood by those of skill in the art that there are other embodiments that are equivalent to the described embodiments. Accordingly, it is to be understood that the invention is not to be limited by the specific illustrated embodiments, but only by the scope of the appended claims.

What is claimed is:

1. A method for generating a function-relation call tree, the method comprising:

acquiring an externally-specified global variable name;

inquiring whether a first function matches the externally-specified global variable name;

in response to the first function matching the externally-specified global variable name, determining whether a code of the first function includes a branch-judgment code;

in response to the code of the first function including the branch-judgment code, dividing the code of the first function to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code;

analyzing the branch-judgment code to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code; and

in response to the related function being called by the code section associated with the attribute value of the branch-judgment code, generating a function-relation call tree between the code section associated with the attribute value and the related function.

2. The method of claim 1 wherein the inquiring whether the first function matches the externally-specified global variable name includes:

performing a static analysis on the code of the first function to acquire a predefined global variable name associated with the first function; and

determining whether the predefined global variable name associated with the first function is the same as the externally-specified global variable name;

wherein:

in response to the predefined global variable name associated with the first function being the same as the externally-specified global variable name, the first function is determined to match the externally-specified global variable name; and

in response to the predefined global variable name associated with the first function being different from the externally-specified global variable

name, the first function is determined not to match the externally-specified global variable name.

3. The method of claim 1 or 2, further comprising:

generating a global function-relation call tree based on at least information associated with merging the function-relation call tree between the code section associated with the attribute value and the related function.

4. The method of claim 3, further comprising:

predefining one or more global variable names and a set of attribute values; and

reading the predefined one or more global variable names and one or more attribute values from the set of attribute values into the first function.

5. A system for generating a function-relation call tree, the system comprising:

an acquisition unit configured to acquire an externally-specified global variable name;

an inquiry unit configured to inquire whether a first function matches the externally-specified global variable name;

a judgment unit configured to determine whether a code of the first function includes a branch-judgment code in response to the first function matching the externally-specified global variable name;

a splitting unit configured to divide the code of the first function to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code in response to the code of the first function including the branch-judgment code;

a determination unit configured to analyze the branch-judgment code to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code; and

a generation unit configured to generate a function-relation call tree between the code section associated with the attribute value and the related function in response to the related function being called by the code section associated with the attribute value of the branch-judgment code.

6. The system of claim 5 wherein the inquiry unit includes:

an analysis sub-unit configured to perform a static analysis on the code of the first function to acquire a predefined global variable name associated with the first function; and

a judgment sub-unit configured to determine whether the predefined global variable name associated with the first function is the same as the externally-specified global variable name;

wherein the judgment sub-unit is further configured to:

in response to the predefined global variable name associated with the first function being the same as the externally-specified global variable name, determine that the first function matches the externally-specified global variable name; and

in response to the predefined global variable name associated with the first function being different from the externally-specified global variable name, determine that the first function does not match the externally-specified global variable name.

7. The system of claim 5 or 6, further comprising:

a merging unit configured to generate a global function-relation call tree based on at least information associated with merging the function-relation call tree between the code section associated with the attribute value and the related function.

8. The system of claim 7, further comprising:

a predefined unit configured to predefine one or more global variable names and a set of attribute values; and.

a reading unit configured to read the predefined one or more global variable names and one or more attribute values from the set of attribute values into the first function.

9. The system of claim 5, further comprising:

one or more data processors; and

a computer-readable storage medium;

wherein one or more of the acquisition unit, the inquiry unit, the judgment unit, the splitting unit, the determination unit, and the generation unit are stored in the storage medium and configured to be executed by the one or more data processors.

10. A non-transitory computer readable storage medium comprising programming instructions for generating a function-relation call tree, the programming instructions configured to cause one or more data processors to execute operations comprising:

acquiring an externally-specified global variable name;

inquiring whether a first function matches the externally-specified global variable name;

in response to the first function matching the externally-specified global variable name, determining whether a code of the first function includes a branch-judgment code;

in response to the code of the first function including the branch-judgment code, dividing the code of the first function to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code;

analyzing the branch-judgment code to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code; and

in response to the related function being called by the code section associated with the attribute value of the branch-judgment code, generating a function-relation call tree between the code section associated with the attribute value and the related function.

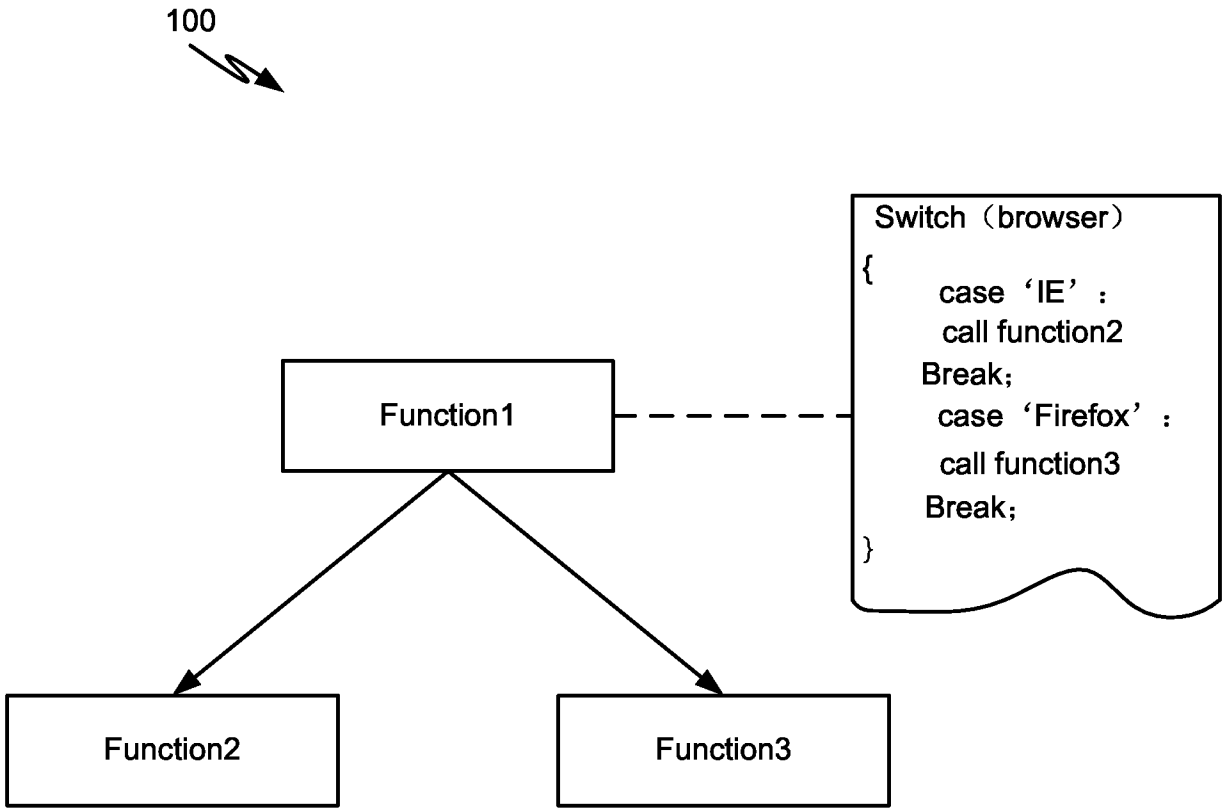

Figure 1
Prior Art

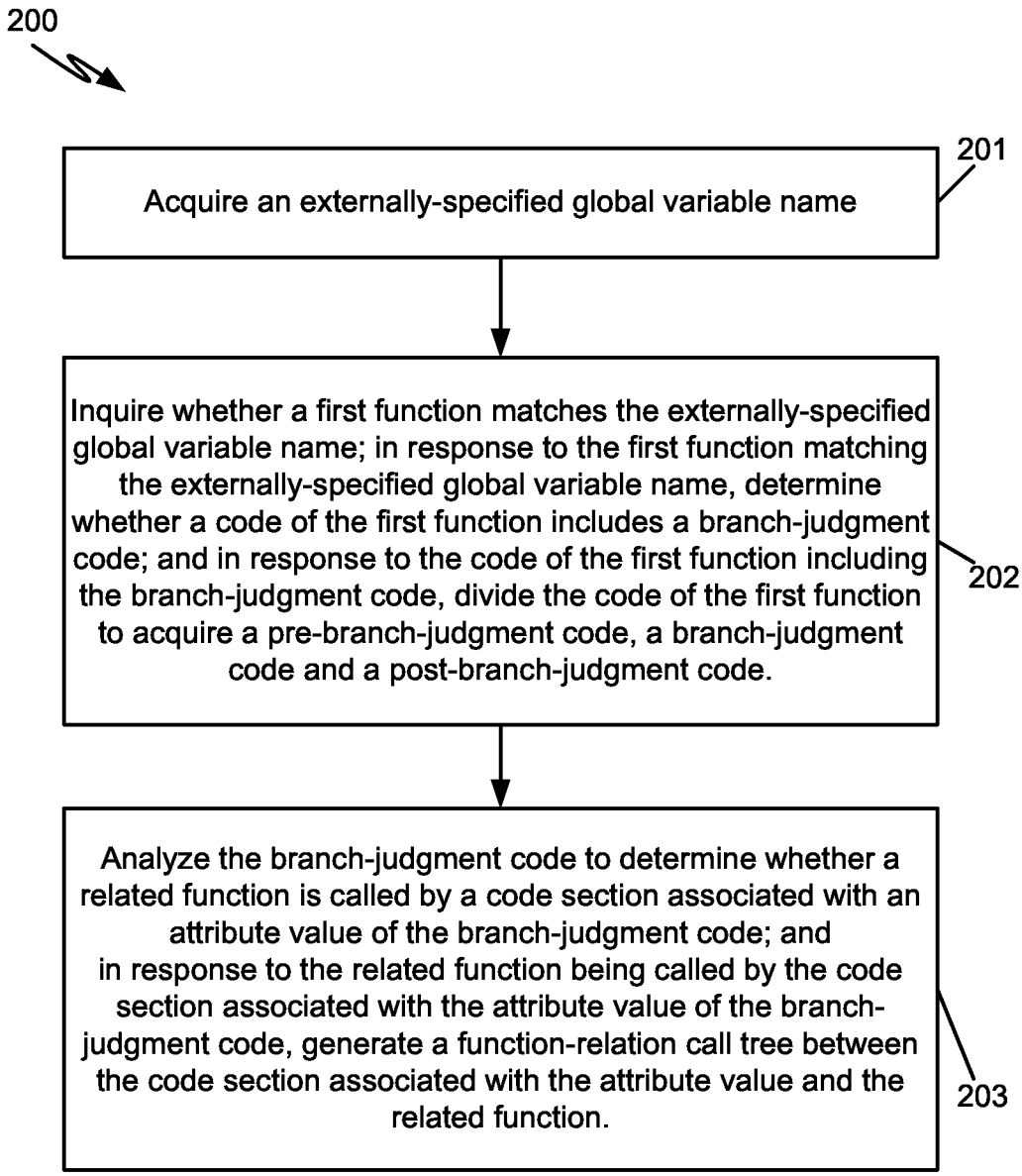
200
Acquire an externally-specified global variable name
201
Inquire whether a first function matches the externally-specified global variable name; in response to the first function matching the externally-specified global variable name, determine whether a code of the first function includes a branch-judgment code; and in response to the code of the first function including the branch-judgment code, divide the code of the first function to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code.
202
Analyze the branch-judgment code to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code; and in response to the related function being called by the code section associated with the attribute value of the branch-judgment code, generate a function-relation call tree between the code section associated with the attribute value and the related function.
203

Figure 2

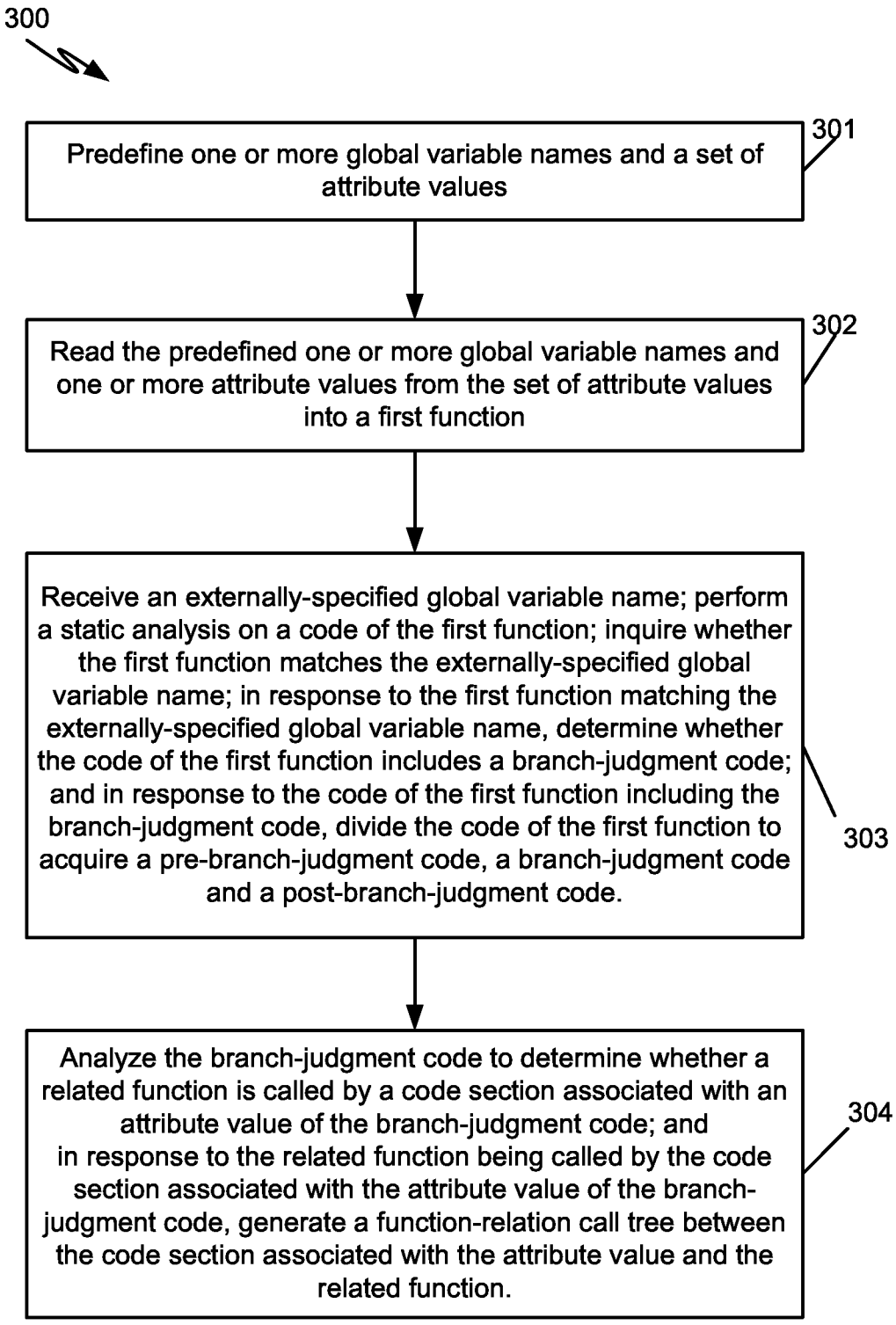
300
Predefine one or more global variable names and a set of attribute values
301
Read the predefined one or more global variable names and one or more attribute values from the set of attribute values into a first function
302
Receive an externally-specified global variable name; perform a static analysis on a code of the first function; inquire whether the first function matches the externally-specified global variable name; in response to the first function matching the externally-specified global variable name, determine whether the code of the first function includes a branch-judgment code; and in response to the code of the first function including the branch-judgment code, divide the code of the first function to acquire a pre-branch-judgment code, a branch-judgment code and a post-branch-judgment code.
303
Analyze the branch-judgment code to determine whether a related function is called by a code section associated with an attribute value of the branch-judgment code; and in response to the related function being called by the code section associated with the attribute value of the branch-judgment code, generate a function-relation call tree between the code section associated with the attribute value and the related function.
304

Figure 3

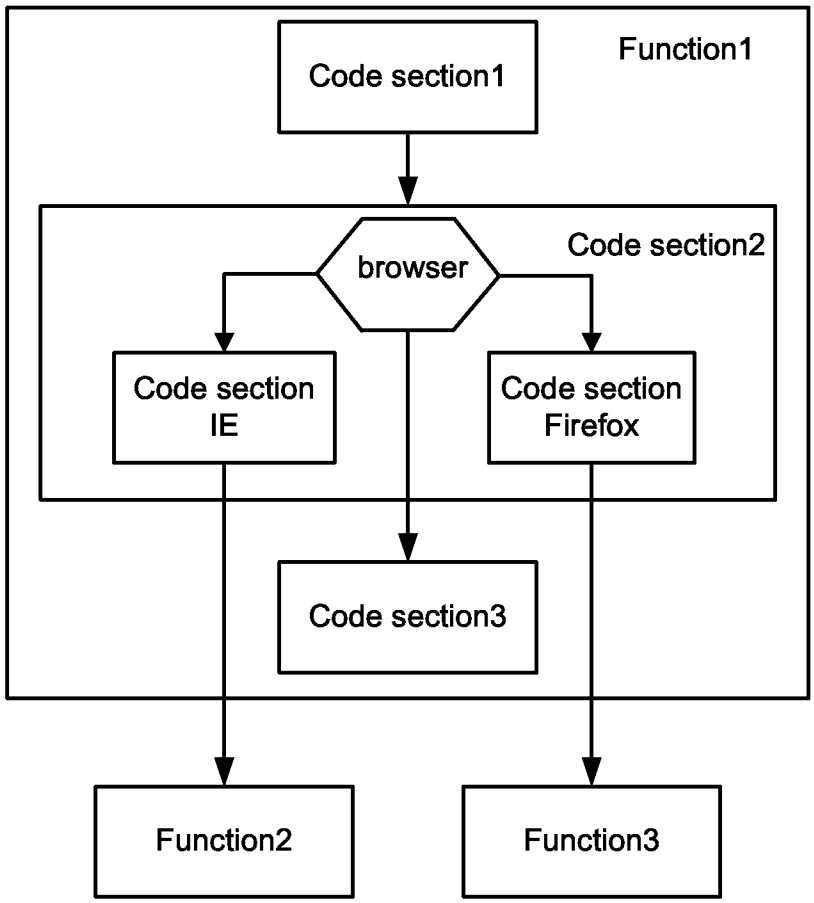
Function1
Code section1
Code section2
browser
Code section
IE
Code section
Firefox
Code section3
Function2
Function3

Figure 4

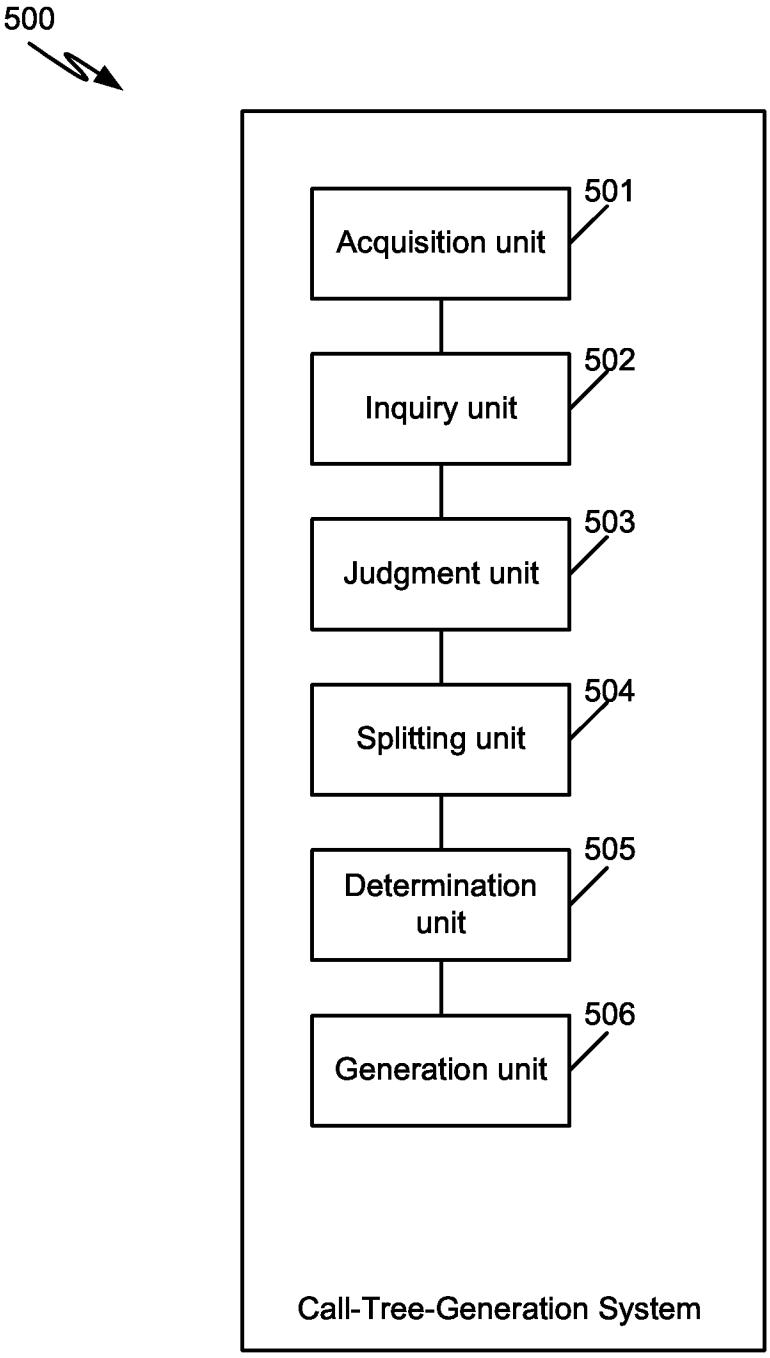
500
501
Acquisition unit
502
Inquiry unit
503
Judgment unit
504
Splitting unit
505
Determination unit
506
Generation unit
Call-Tree-Generation System

Figure 5

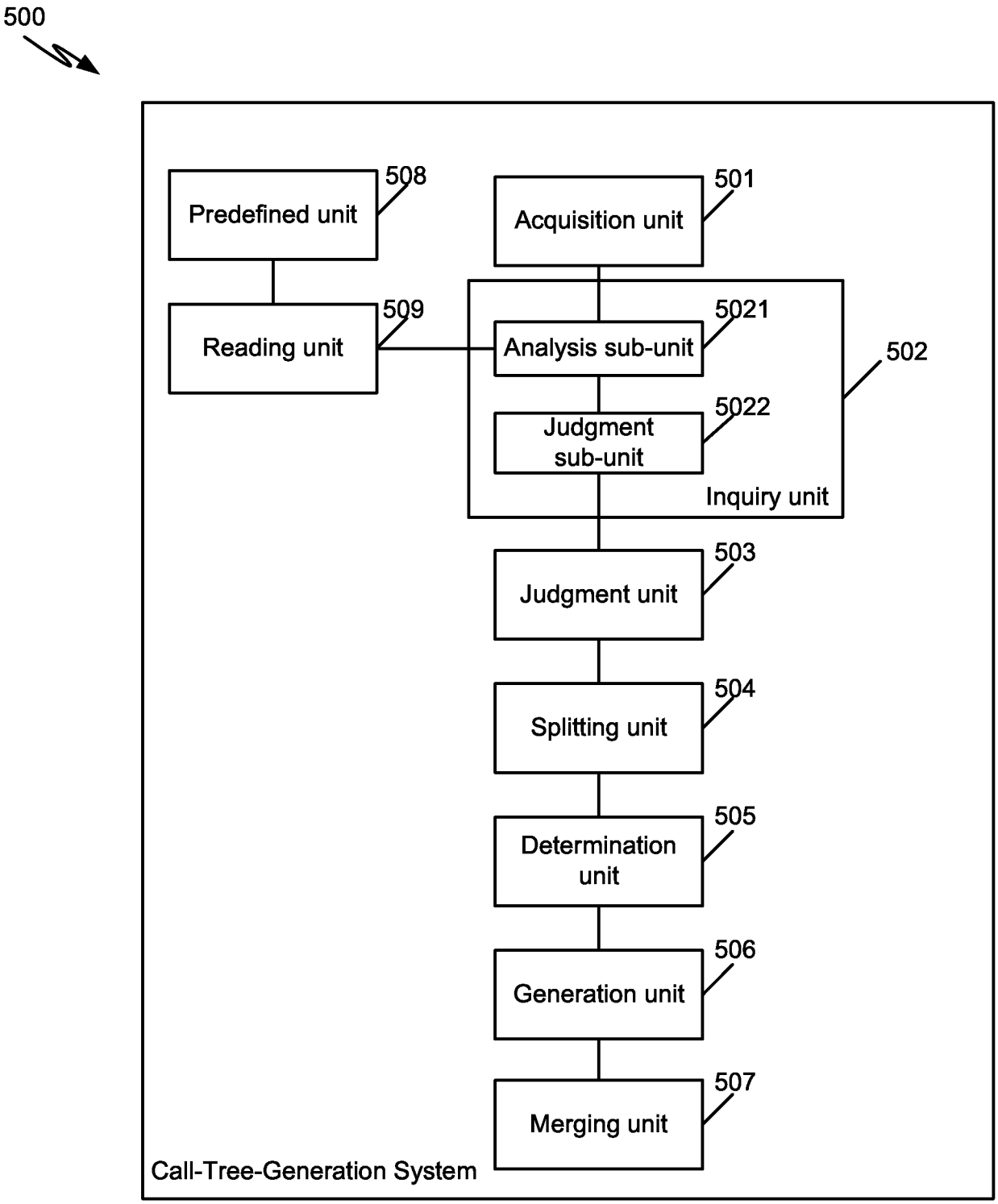
500
Predefined unit
508
Acquisition unit
501
Reading unit
509
Analysis sub-unit
5021
502
Judgment sub-unit
5022
Inquiry unit
Judgment unit
503
Splitting unit
504
Determination unit
505
Generation unit
506
Merging unit
507
Call-Tree-Generation System

Figure 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2013/087380

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/44 (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:G06F9/-;G06F11/-;G06F17/-

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)

WPI, EPDOC, CNABS, CNKI: program, software, subroutine, function, call, tree, relation, graph, condition, conditional, select????, judgment, sentence, statement, branch

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP 2002215391 A (FUJITSU LTD) 02 Aug. 2002 (02.08.2002) paragraphs [0018]-[0042] in the description, figures 1-9	1-10
A	CN 1908892 A (WANG TONG) 07 Feb. 2007 (07.02.2007) the whole document	1-10
A	KR 20010075872 A (KOREA ELECTRONICS&TELECOM RES INST) 11 Aug. 2001 (11.08.2001) the whole document	1-10
A	CN 101286119 A (HUAYAO HUANYU TECHNOLOGY BEIJING CO LTD) 15 Oct. 2008 (15.10.2008) the whole document	1-10

☐ 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
05 Jan. 2014 (05.01.2014)	**06 Mar. 2014 (06.03.2014)**
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 HAO, Xiaoli Telephone No. (86-10)62411770

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

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2013/087380

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
JP 2002215391 A	02.08.2002	None	
CN 1908892 A	07.02.2007	WO 2007014521 A1	08.02.2007
KR 20010075872 A	11.08.2001	KR 100333670 B	22.04.2002
CN 101286119 A	15.10.2008	CN 100576172 C	30.12.2009