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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 December 2014 (24.12.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/201860 A1

(51) International Patent Classification:
G06F 3/0481 (2013.01)

(21) International Application Number:
PCT/CN2014/070098

(22) International Filing Date:
3 January 2014 (03.01.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
201310247260.4 20 June 2013 (20.06.2013) CN

(71) Applicant: **TENCENT TECHNOLOGY (SHENZHEN) COMPANY LIMITED** [CN/CN]; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518044 (CN).

(72) Inventor: **SHU, Yuqiang**; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518044 (CN).

(74) Agent: **ADVANCE CHINA IP LAW OFFICE**; Room 3901, No.85 Huacheng Avenue, Tianhe District, Guangzhou, Guangdong 510623 (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**: METHOD AND DEVICE OF CONTROLLING INPUT FOR INPUT BOXES

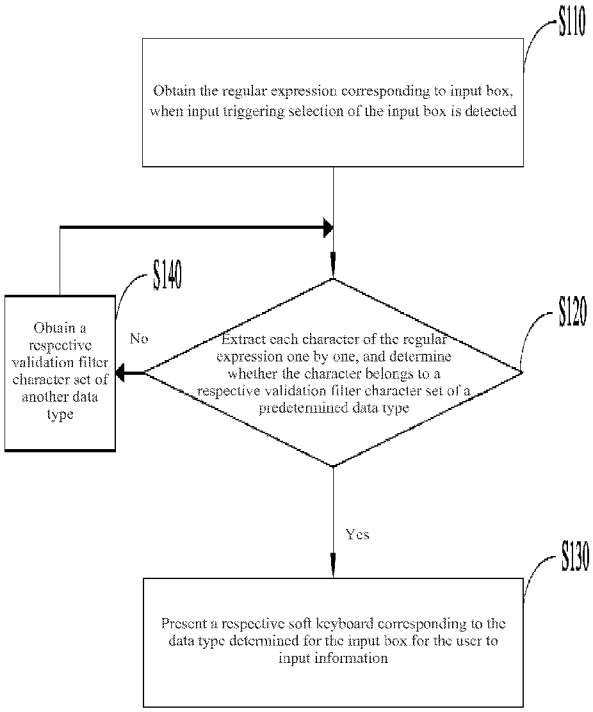


Fig. 1

(57) **Abstract**: A method and device for controlling information input for text boxes are disclosed. The method includes: at a device having one or more processors, memory, and a display: detecting selection of a text input box in a user interface; in response to detecting the selection of the text input box and prior to receiving input text for the text input box, obtaining a regular expression for acceptable input corresponding to text input box; for each character in the regular expression, determining whether the character belongs to a respective validation filter character set for a predetermined data type; and in response to a determination that all characters in the regular expression belongs to the respective validation filter character set for the predetermined data type, selectively presenting a respective character keyboard corresponding to the predetermined data type to a user for text entry.

WO 2014/201860 A1

METHOD AND DEVICE OF CONTROLLING INPUT FOR INPUT BOXES

RELATED APPLICATIONS

[0001] This application claims priority to Chinese Patent Application Serial No. CN201310247260.4, entitled "Method and Device of Controlling Information Input for Input Boxes", filed June 20, 2013, the entirety of which is incorporated herein by reference.

FIELD OF THE INVENTION

[0002] The present disclosure relates to the information input field, especially relates to a method and device for controlling information input for input boxes.

BACKGROUND OF THE INVENTION

[0003] In a web page, input boxes are often provided for users to input words or textual information, email addresses, telephone numbers, or the hyperlink information of files, and so on. During the design of these input boxes, web developers often prescribe the data types acceptable for each input box, for example, numeric type, Chinese character type, Roman letter type, and so on. After the user has provided the data input into the input box and submitted the data input, e.g., by pressing "enter" or moving onto the next input box, the inputted data is verified against the data type(s) prescribed for the input box to determine whether the input data meets the requirements of the input box. Frequently, the requirement of the input box is provided in the form of a regular expression corresponding to the input box.

SUMMARY

[0004] As described in the background section, different input boxes may require input of different character types (e.g., numbers, symbols, Roman characters, Chinese characters, Greek characters, etc.). Some input boxes may require a single character type or a mixed character types following respective formatting requirements. For example, an input box for receiving a telephone number may require all input characters to be numbers, and requires that the total number of numbers in the input box be exactly 10. For another example, an input box for receiving a first name may require all input characters to be Roman or Chinese characters, and for input containing Roman characters, no space character is permitted in the input. Other character type and format requirements for different information types are possible. One problem with inputting information in the input box is that, the user often has to switch between keyboards (e.g., the number keyboard, English character keyboard, symbols keyboard, and Chinese Pinyin keyboard) in order to complete input to an input box. This is very cumbersome and inefficient.

[0005] As described in the present disclosure, a method and device for controlling information input for input boxes makes the determination of the required data type for a currently selected input box, and based on the required character type, proactively and selectively presenting a

suitable keyboard to the user according to the current input needs of the user. Since a regular expression for post-entry validation of user input is frequently provide with the input boxes on web pages, the present method and device takes advantage of this available regular expression to determine the required data type and hence the suitable keyboard for presentation to the user before the user starts the character entry into the input box. In some embodiments, only if all characters required in an input box can be found in a given keyboard, will the keyboard be automatically presented in lieu of a current or default keyboard. In some embodiments, the keyboard corresponding to the initial character input is provided first.

[0006] In some embodiments, a method of input control for text input boxes, comprising: at a device having one or more processors, memory, and a display: detecting selection of a text input box in a user interface; in response to detecting the selection of the text input box and prior to receiving input text for the text input box, obtaining a regular expression for acceptable input corresponding to text input box; for each character in the regular expression, determining whether the character belongs to a respective validation filter character set for a predetermined data type; and in response to a determination that all characters in the regular expression belongs to the respective validation filter character set for the predetermined data type, selectively presenting a respective character keyboard corresponding to the predetermined data type to a user for text entry. In this way, when a user needs to input information in the input box, the user may need to continuously switch keyboard due to the data type of different input boxes, which will reduce the efficiency of information input.

[0007] In another aspect, a device comprises one or more processors, memory, and one or more program modules stored in the memory and configured for execution by the one or more processors. The one or more program modules include instructions for performing the method described above. In another aspect, a non-transitory computer readable storage medium having stored thereon instructions, which, when executed by a device, cause the device to perform the method described above.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The aforementioned features and advantages of the invention as well as additional features and advantages thereof will be more clearly understood hereinafter as a result of a detailed description of preferred embodiments when taken in conjunction with the drawings.

[0009] Fig. 1 is the schematic flowchart diagram of an information input control method for input boxes in accordance with some embodiments;

[0010] Fig. 2 is the schematic flowchart diagram an information input control method for input boxes in accordance with some embodiments;

[0011] Fig. 3 is the schematic flowchart diagram an information input control method for input boxes in accordance with some embodiments;

[0012] Fig. 4 is the schematic flowchart diagram of an information input control method for input boxes in accordance with some embodiments;

[0013] Fig. 5 is the schematic flowchart diagram of an information input control method for input boxes in accordance with some embodiments;

[0014] Fig. 6a is the schematic flowchart diagram of processing a character when the character falls between "{ }", in an information input control method for input boxes in accordance with some embodiments;

[0015] Fig. 6b is the schematic flowchart diagram of processing a character when the character falls between "( )", in an information input control method for input boxes in accordance with some embodiments;

[0016] Fig. 6c is the schematic flowchart diagram of processing a character when the character follows "^" or falls between "[ ]", in an information input control method for input boxes in accordance with some embodiments;

[0017] Fig. 7 is the function module schematic diagram of an information input control apparatus for input boxes in accordance with some embodiments; and

[0018] Fig. 8 is a block diagram of a device for controlling information input for input boxes in accordance with some embodiments.

[0019] Like reference numerals refer to corresponding parts throughout the several views of the drawings.

DESCRIPTION OF EMBODIMENTS

[0020] The following describes further technical scheme of the present invention in combination of attached drawings of specification as well as specific embodiments. It should be understood that the specific embodiments described herein is merely used to illustrate the present invention, rather than to limit the present invention.

[0021] In some embodiments, when an input box in a user interface (e.g., a fill-able form on a web page) has been selected and awaits information input (e.g., text input through keyboard entry) from a user, the user device automatically presents the soft keyboard (e.g., a soft English character keyboard, number keyboard, Pinyin keyboard, or symbols keyboard, etc. on a touch-sensitive display screen), according to the type of input data required by the input box. Therefore, manual switching soft keyboards repeatedly during the information input by the user can be reduced or avoided, which improves the efficiency of information input on the user device.

[0022] The description of the following embodiments uses the information input control of an input box on webpage presented in a browser as an example. However, this is not meant to limit the scope of application and protection of the methods described herein. The method of automatically, proactively selecting and presenting a suitable soft keyboard on a touch-sensitive display screen based on the required data type of a currently selected text input box can be applied to input boxes

that appear in other types of interfaces, e.g., various applications, information collection devices, automated teller machines, etc. The utilization of the regular expression for post-entry data validation may be replaced by the utilization of other types of format requirement information associated with the input boxes which may be available to the user device displaying the input box.

[0023] Fig. 1 illustrates a method of controlling information input for input boxes in accordance with some embodiments. The method is performed on a user device having one or more processors, memory, and a touch-sensitive display screen for presenting one or more input boxes and selectively presenting one of multiple available soft keyboards in response to user selection or the keyboard selection method described herein. The information input control method for input boxes includes:

[0024] Step S110, the device obtains the regular expression corresponding to input box, when input triggering selection of the input box is detected;

[0025] Specifically, the device detects selection of a text input box in a user interface; and in response to detecting the selection of the text input box and prior to receiving input text for the text input box, the device obtains a regular expression for acceptable input corresponding to text input box.

[0026] Taking a webpage displayed in a browser of a user device as an example, the page information for the webpage in the browser being expressed in the form of HTML15 code. If there is an input box in the web page, the attribute of the input box is set in the HTML15 code. For example, a "max" attribute is used to specify the maximum value of the input field, the "min" attribute is used to specify the minimum value of the input field, the "name" attribute is used to specify the name of input element, and the "pattern" attribute is used to specify the pattern or the format of the value of the input field. Specifically, the content of the "pattern" attribute is in the form of a regular expression. For example, the regular expression in for an input box may be "[0 - 9]", which means that the input value for the input box must be an integer between 0 - 9. The detection of the input triggering selection of the input box can include: the device detects whether there is any input box in the webpage browsed by the user; if there is an input box, the device obtains movement of the cursor (or the contact point of a touch input on a touch-sensitive display screen) on the interface; if the cursor (or contact point) switches input focus onto the input box, the focus is locked onto the input box (i.e., the input box is selected), and the device generates an input triggering command; then, according to this input triggering command, the device obtains the regular expression in the "pattern" attribute corresponding to the selected input box.

[0027] Step S120, the device extracts each character of the regular expression one by one, and determines, for each character in the regular expression, whether the character belongs to a respective validation filter character set of a predetermined data type;

[0028] In response to a determination that all characters in the regular expression belong to the respective validation filter character set of the predetermined data type, the device proceeds to Step S130; and in response to a determination that any one character in the regular expression does not belong to the validation filter character set of the predetermined data type, the device proceeds to Step S140;

[0029] In some embodiments, the predetermined data type is one of three kinds of data types, namely, Roman letters, Chinese characters, and numbers. The regular expression for each kind of data type is different, and the regular expression corresponding to same data type can be different as well due to special requirements expressed in the regular expression. In some embodiments, the regular patterns of various regular expressions corresponding to a given data type can be observed, and the validation filter character set of the given data type can be established accordingly. Taking the case of the validation filter character set for the number type as an example, in some embodiments, the device matches each character in the obtained regular expression corresponding to the selected input box with the validation filter character set of the number type, so as to determine the input data type required by the selected input box .

[0030] Step S130, the device presents a respective soft keyboard corresponding to the data type determined for the input box for the user to input information;

[0031] Specifically, in some embodiments, in response to a determination that all characters in the regular expression belongs to the respective validation filter character set for the predetermined data type, the device presents a respective character keyboard corresponding to the predetermined data type to a user for text entry.

[0032] For example, if each character in the obtained regular expression of the input box belongs to the validation filter character set of the number type, it means that the data type required by the input box is the number type, and consequently, the device presents the numeric soft keyboard to the user.

[0033] Step S140, the device obtains a respective validation filter character set of another data type, and returns to Step S120, so as to determine whether the data type required by the selected input box is this other data type. In some embodiments, a respective validation filter character set is established for each soft keyboard. In some embodiments, a respective soft keyboard can be pre-established for each predetermined data type, and which includes all possible keys that may be used for the data type. In such embodiments, the respective soft keyboard for the determined data type of the input box can be presented once the data type of the input box has been determined. In some embodiments, the data type of the input box may be defined according to specific needs, and may not be limited to the number keyboard, the Roman letter keyboard, and the Pinyin keyboard. Data types each defined by a respective validation filter character set including some or all of multiple conventional keyboards are possible in various embodiments.

[0034] In some embodiments, if any one character in the regular expression does not belong to the validation filter character set of the predetermined data type, the device ends the process. In some embodiments, the device can also return to Step S120 again, and perform the process for a next predetermined data type: e.g., extracting each character of the regular expression one by one, and determining whether the character belongs to the validation filter character set of the next data type. For example, when the device determines that a character in the regular expression does not belong to the validation filter character set of the number type, the device determines again whether each character of the regular expression belongs to the preset filtering character validation set of the Chinese character type. The process can continue for other predetermined data types, until the data type of the input box has been identified.

[0035] In some embodiments, when an input box is selected for information input, the device, according to the regular expression corresponding to the input box, determines the data type required by the input box, so as to present the corresponding soft keyboard of the determined data type for user to input information in the input box. As a result, the user does not need to frequently manually switch the soft keyboards for inputting information in the input box, which improves the efficiency of information input on the device.

[0036] In some embodiments, as shown in Fig. 2, a method of controlling information input for input boxes further include the following steps. Specifically, the validation filter character set includes regular validation filter characters (e.g., in a subset B) and special validation filter characters (e.g., in a subset C). For example, in some embodiments, the subset (B) of regular validation filter characters includes {*, +, $, |, \b, \d, \B, \num}, and the subset (C) of special validation filter characters includes {^, [], {}, (), \nm}. In some embodiments, the aforementioned Step S120 includes:

[0037] Step S121, the device sets the current value of the index for the current character in the regular expression, where the initial index value is 0;

[0038] First, the index value is set to zero, in other words, the processing starts from the first character of the regular expression. After the determination for a current character is completed, the index value is updated by adding 1 to the current index value.

[0039] Step S122, the device determines whether the index value has reached the string length (e.g., "strlen") of the regular expression; if no, proceed to Step S123; otherwise, end the process;

[0040] When the current index value equals to the string length of the regular expression, it means that the determination for all characters in the regular expression have been completed.

[0041] Step S123, the device obtains the character corresponding to the current index value as the current character to be processed;

[0042] In some embodiments, through c=str[index], the device can extract each character of the regular expression one by one, as the current character to be processed.

[0043] Step S124, the device determines whether the current character belongs to the subset B of regular validation filter characters; if yes, the device proceeds to Step S125, otherwise, the device proceeds to Step S126;

[0044] Step S125, the device adds 1 to the current index value, and returns to S122;

[0045] In some embodiments, when the current character *c* belongs to the subset (B) of regular validation filter characters, the device adds 1 to the current index value, and then returns to Step S122. For example, in some embodiments, the device obtains the next character by *c*=str[index] of the regular expression, and uses the obtained character as the current character.

[0046] Step S126, the device determines whether the current character belongs to the subset (C) of special validation filter characters; if so, the device proceeds to Step S127, otherwise, the device ends the process;

[0047] Step S127, the device obtains the characters between the current character (e.g., "(", or "{") and the counterpart character (e.g., ")" or "}") corresponding to the current character as a character subset, and processes the character subset accordingly.

[0048] Specifically, n some embodiments, when the current character belongs to the subset (C) of special validation filter characters, the device obtains all characters that fall between the current character and its matching counterpart character, and treats these characters that fall between the two matching special validation filter characters as a character subset, and then conducts corresponding special processing for the character subset depending on what the two special filter characters are. For example, in some embodiments, for the character subset between a pair of special validation filter characters "{" and "}", the device directly passes the characters in the character subset without further processing thereof; for the character subset that fall between a pair of special validation filter characters "(" and ")" or a pair of special validation filter characters "[" and "]", the device makes individual determination for each character within this character subset, and determines whether all of them belong to the validation filter character set of the predetermined data type.

[0049] Specifically, in some embodiments, for each character in the regular expression, the process for determining whether the character belongs to the respective validation filter character set for the predetermined data type further includes: determining whether the character belongs to the subset of regular validation filter characters or the subset of special validation filter characters; in accordance with a determination that the character belongs to the subset of regular validation filter characters, proceeding to process a next character in the regular expression; and in accordance with a determination that the character belongs to the subset of special validation filter characters,

processing all characters that fall between the character and a counterpart character of the character in the regular expression as a respective character subset.

[0050] In some embodiments, as shown in Fig. 3, the method for controlling information input for input boxes further include the following sub-steps. In some embodiments, after the aforementioned Step S123, the information input control method also includes:

[0051] Step S128, the device determines whether the current character belongs to a subset (A) of stop characters for the predetermined data type; if yes, the device ends the process, otherwise, the device proceeds to Step S124;

[0052] In some embodiments, in order to reduce the number of determinations made for a given data type, a set of stop characters can be pre-established for the given data type. Taking the number type as an example, its stop character set (e.g., a subset A) includes {., /, \D, \w, \W, \cx, \f, \n, \r, \s, \t, \v, letters, Chinese characters, \xn}. The meaning of these characters (including the ones with a "\" on the left are well known to a person of ordinary skills in the art, and is not repeated here for brevity. Before determining whether the current character belongs to the validation filter character set of the predetermined data type, the device determines whether the current character belongs to the set of stop characters for the data type; if the current character belongs to the set of stop characters, the device can directly determine that the data type of this input box is not the predetermined data type (e.g., the number type). Therefore, through pre-establishing the respective stop character set for each predetermined data type, and conducting the above preliminary determination for the current character, the subsequent determination process can be eliminated if the preliminary determination determines that the predetermined data type is not the data type of the input box.

[0053] Specifically, in some embodiments, for each character in the regular expression: when determining whether the character belongs to the subset of regular validation filter characters or the subset of special validation filter characters, the device first determines whether the character belongs to a stop character set for the predetermined data type; and in accordance with a determination that the character does not belong to the stop character set for the predetermined data type, the device proceeds to determine whether the character belongs to the subset of regular validation filter characters or the subset of special validation filter characters; otherwise, the device moves on to process the regular expression for a next predetermined data type.

[0054] In some embodiments, as shown in Fig. 4, the information input control method for input boxes further include the following sub-steps. In some embodiments, some stop characters in the set of stop characters for the predetermined data type includes characters each having a respective backslash character. Thus, in order to ease the processing for the characters, the set (A) of stop characters are divided into a subset (A1) of stop characters each having a backslash, and a subset (A2) of stop character without a backslash. In some embodiments, the mentioned set of stop characters (A)

includes a subset of stop characters A1 having a backslash condition, and a subset of stop characters A2 without a backslash condition. As such, in some embodiments, the aforementioned Step S128 includes:

[0055] Step S1281, the device determines whether the backslash attribute of the current character is "true"; if yes, then the device proceeds to Step S1282, otherwise, the device proceeds to Step S1283;

[0056] In some embodiments, the device also sets a backslash attribute "isBackSlant" as "TRUE" to indicate the presence of the backslash in the current character in Step S121. The backslash attribute "isBackSlant" is "FALSE" by default indicating that no backslash is present in the current character.

[0057] Step S1282, the device determines whether the current character *c* belongs to the subset (A1) of stop characters having a backslash condition (i.e., having a backslash in the character);

[0058] Specifically, the device obtains the subset (A1) of stop characters with a backslash condition, and then determines whether the current character matches with any stop characters in the subset (A1) of stop characters with a backslash condition. If there is a match, the device ends the process; otherwise, the device proceeds to Step S124.

[0059] Step S1283, the device determines whether the current character is a backslash "\"; if yes, then the device proceeds to Step S1284, otherwise, the device proceeds to Step S1285;

[0060] Specifically, in order to avoid misjudgment in a scenario in which "\" serves as the first character of the regular expression, when the "isBackSlant" attribute value is determined to be "FALSE", the device shall determine whether the current character is a backslash "\".

[0061] Step S1284, the device adds 1 to the current index value, and sets the attribute value "isBackSlant" as "TRUE", and then returns to Step S122;

[0062] When the current character *c* is determined to be a backslash "\", the device sets the "isBackSlant" attribute value to "TRUE", adds 1 to the current index value, and then returns to Step S122.

[0063] Step S1285, the device determines whether the current character *c* belongs to the subset (A2) of stop characters without a backslash condition.

[0064] Specifically, the device obtains the subset (A2) of stop characters without a backslash condition, and then determines whether the current character matches with any character in the subset (A2) of stop characters without a backslash condition. If there is a match, the device ends the process; otherwise, the device proceeds to Step S124.

[0065] Specifically, in some embodiments, determining whether the character belongs to a stop character set for the predetermined data type further includes: determining whether the character has an associated backslash; in accordance with a determination that the character has an associated

backslash, determining whether the character belongs to the subset of stop characters having a backslash condition; and in accordance with a determination that the character does not have any associated backslash, determining whether the character belongs to the subset of stop characters without any backslash condition.

[0066] Furthermore, in some embodiments, as shown in Fig. 5, the method of controlling information input for input boxes further includes the following sub-steps. As some characters for the predetermined data type includes characters each having a respective backslash character. Thus, in order to ease the processing for the characters, in some embodiments, the subset (B) of regular validation filter characters and the subset (C) of special validation filter characters may also include characters each having an associated backslash condition. In some embodiments, the subset (B) of regular validation filter characters includes a subset (B1) of regular validation filter character with a backslash condition, and a subset (B2) of regular validation filter characters without a backslash condition. In some embodiments, the subset (C) of special validation filter characters include a subset (C1) of special validation filter characters with a backslash condition and a subset (C2) of special validation filter characters without a backslash condition.

[0067] In some embodiments, the process further include the following steps:

[0068] Step S1241, when the device has determined that the current character *c* does not belong to the subset (A1) of stop characters each having a backslash, the device also determines whether the current character *c* belongs to the subset (B1) of regular validation filter characters with a backslash; if yes, the device proceeds to Step S1263, or otherwise, the device proceeds to Step S1261;

[0069] Step S1242, when the device has determined that the current character *c* does not belong to the subset (A2) of stop characters without a backslash, the device also determines whether the current character *c* belongs to the subset (B2) of regular validation filter characters without a backslash; if yes, the device proceeds to Step S1263, otherwise, the device proceeds to Step S1262;

[0070] Step S1261, the device determines whether the current character *c* belongs to the subset (C1) of special validation filter characters having a backslash; if yes, the device proceeds to Step S1263, otherwise, the device ends the process;

[0071] Step S1262, the device determines whether the current character *c* belongs to the subset (C2) of special validation filter characters without a backslash; if yes, the device proceeds to Step S127, otherwise, the device ends the process;

[0072] Step S1263, the device adds 1 to the current index value, and proceeds to S122.

[0073] It can be understood that, in some embodiments, when determining whether the data type of the input box is of the number type only by user filter characters, it is possible to, after extracting each character of the regular expression one by one, determine whether a backslash exists in the current character by reading the value of the backslash attribute. If backslash exists in the

current character, then the current character *c* is matched against the characters in subsets A1 and C1; if backslash does not exist in the current character, then the current character c is matched against the characters in A2 and C2. Details of the matching process are disclosed in more details above.

[0074] In some embodiments, as shown in Fig. 6a, Fig. 6b and Fig. 6c, the method for controlling information input for input boxes further includes the following sub-steps. These sub-steps involve mainly the special processing of a character *c* belonging to the subset (C2) of special validation filter characters without a backslash condition. In some embodiments, C2 includes {^, [], {}, ()}. The following will give an explanation of special processing of the character *c* when *c* belongs to C2.

[0075] As shown in Fig. 6a, the current character *c* is "{". As all characters included in "{}" as well as "{" and "}" are filtered, when c is determined to be "{", the device adds 1 to the current index value, and determines whether the next character is "}", if it is, then the device ends this special processing, and, after adding 1 to the current index value, the device returns to Step S122. If it is not, then the device adds 1 to the current index value, and returns to determine whether the next character is "}".

[0076] As shown in Fig. 6b, the current character c is "(". As all characters included in "( )" need recursive processing, when c is determined to be "(", the device first defines a new variable "oldIndex" for saving the current index value; then adds 1 to the index current index value, and determines whether the character corresponding to the current index value is ")", if yes, the device obtains all characters included within [oldIndex, index] are obtained as a character subset, otherwise, the device adds 1 to the index value and determine again whether the character corresponding to the current index value is ")", and so on. In the end, the device re-determines whether each of the characters in the character subset belongs to the validation filter character set of the predetermined data type; if yes, then the device ends the special processing, adds 1 to index, and, after which, returns to Step S122. If any one of characters in the character subset does not belong to the validation filter character set of the predetermined data type, then the device ends the entire process.

[0077] As shown in Fig. 6c, if the current character *c* is "^". As "^" will be filtered when serving as the first character of the regular expression, if the current character *c* is determined to be "^", the device first defines a new variable "oldIndex" for saving the current index value; then the device determines whether the current index value is 0; if yes, the device adds 1 to the current index value, and, after which, returns to Step S122, otherwise, the device determines whether the character before the current character *c* is "[". When the character before the current character *c* is determined to be "[", the device obtains all characters between "[" and "]" as the character subset. Then the device re-determines whether each of the characters in the character subset belongs to the validation filter character set of the predetermined data type; if yes, then the device ends the special processing, adds 1 to the current index value, and, after which, returns to Step S122; if any one of the characters

in the character subset does not belong to the validation filter character set of the predetermined data type, then the device ends the entire process.

[0078] If the current character *c* is "[", the device obtains characters between "[" and "]" as the character subset. Then the device re-determines whether each of the characters in the character subset belongs to the validation filter character set of the predetermined data type; if yes, then the device ends the special processing, adds 1 to the current index value, and, after which, returns to Step S122; if any one of the characters in the character subset does not belong to the validation filter character set of the predetermined data type, then the device ends the entire process.

[0079] In aforementioned embodiments, the numbers type is used as an example data type of the input box. Naturally, input boxes having other data types are possible. The specific determination rules and the validation filter character set for the different data types may be different. For example, in some embodiments, when making the determination for the Chinese character data type, the set of stop characters A is {./,\D,\w,\W,\cx,\f,\n,\r,\s,\t,\v, letters,\xn,\b,\d,\B,\num}, the subset (B) of regular validation filter characters is {*,+,$,|}, and the subset (C) of special validation filter characters is {^,[],{},(),\nm,\u}. In another example, in some embodiments, when making the determination for the English letters data types, the set of stop characters A is {.,/,\D,\w,\W,\cx,\f,\n,\r,\s,\t,\v, Chinese,\xn,\b,\d,\B,\num}, the subset (B) of regular validation filter characters is {*,+,$,|,A-Z,a-z}, and the subset (C) of special validation filter characters is {^,[],{},(),\nm}.

[0080] Fig. 7 shows an example device for controlling information input for input boxes in accordance with some embodiments. The example device performs the methods described above with respect to Figs. 1-6c. In some embodiments, the device includes:

[0081] Information input detection module 110, configured to obtain the regular expression corresponding to the currently selected input box, when the input triggering command of the input box is detected;

[0082] Input box type matching module 120, configured to extract each character of the regular expression one by one, and determine whether each character belongs to the validation filter character set of the predetermined data type;

[0083] Processing module 130, configured to selectively present the soft keyboard corresponding to the predetermined data type for the user to input information, when all characters belong to the validation filter character set of the predetermined data type.

[0084] Furthermore, the mentioned validation filter character set includes a subset of regular validation filter characters and a subset of special validation filter characters; and the mentioned input box type matching module 120 is used to:

[0085] Extract each character of the regular expression one by one as the current character, and determine whether the current character belongs to the subset of regular validation filter characters or the subset of special validation filter characters;

[0086] When the current character belongs to the subset of regular validation filter characters, then take the next character as the current character;

[0087] When the current character belongs to the subset of special validation filter characters, then, obtain the characters between the current character and its matching character as the character subset, and process the character subset accordingly.

[0088] Furthermore, the mentioned input box type matching module 120 can also be used to:

[0089] Determine whether the current character belongs to a set of stop character for the predetermined data type;

[0090] When the current character does not belong to the set of stop characters for the data type, then, determine whether the current character belongs to the subset of regular validation filter characters or the subset of special validation filter characters.

[0091] Furthermore, the set of stop characters include a subset of stop characters with a backslash condition and a subset of stop characters without a backslash condition. The mentioned input box type matching module 120 can also be used to:

[0092] Determine whether a backslash exists in the current character based on the backslash attribute of the character;

[0093] If a backslash is determined to exist in the current character, then determine whether the current character belongs to the subset of stop characters with a backslash condition;

[0094] If no backslash is determined to exist in the current character, then determine whether the current character belongs to the subset of stop characters without a backslash condition.

[0095] Furthermore, the mentioned validation filter character set includes a subset of validation filter characters with a backslash condition and a subset of validation filter characters without a backslash condition. The mentioned input box type matching module 120 can also be used to:

[0096] If a backslash is determined to exist in the current character, then determine whether the current character belongs to the subset of validation filter characters with a backslash condition;

[0097] If no backslash is determined to exist in the current character, then determine whether the current character belongs to the subset of validation filter characters without a backslash condition.

[0098] Furthermore, the mentioned processing module can also be used to: when the current character does not belong to the validation filter character set of the predetermined data type, then obtain the preset filtering character validation set of the next data type; when each character of regular expression belongs to the preset filtering character validation set of the next data type by

judgment, then pop up the soft keyboard corresponding to this next data type for user to input information;

[0099] The mentioned input box type matching module is also configured to extract each character of the regular expression one by one, and determine whether each character belongs to the validation filter character set of the predetermined data type.

[00100] Refer to the previous method embodiments for working principle and its realized functions of each functional module in the aforementioned control device, and redundant description will be omitted herein.

[00101] Fig. 8 is a block diagram of a device 800 for performing the methods of controlling information input for input boxes described above in accordance with some embodiments. In some embodiments, the device is a terminal device (e.g., a mobile telephone, a tablet device, a portable computer) that displays one or more input boxes in a graphical user interface, and corresponding soft keyboards on the display screen. The user may select keys on the soft keyboard by a touch input, or a cursor or voice-based selection input directed to the respective keys.

[00102] In some embodiments, device 800, typically, includes one or more processing units (CPUs) 802, one or more network interfaces 804, memory 806, and one or more communication buses 808 for interconnecting these components (sometimes called a chipset). Client device 800 also includes a user interface 810. User interface 810 includes one or more output devices 812 that enable presentation of media content. User interface 810 also includes one or more input devices 814, including user interface components that facilitate user input such as a keyboard, keypad, a mouse, a voice-command input unit or microphone, a touch screen display, a touch-sensitive input pad, a gesture capturing camera, or other input buttons or controls. Furthermore, some client devices 800 use a microphone and voice recognition or a camera and gesture recognition to supplement or replace the keyboard. Memory 806 includes high-speed random access memory, such as DRAM, SRAM, DDR RAM, or other random access solid state memory devices; and, optionally, includes non-volatile memory, such as one or more magnetic disk storage devices, optical disk storage devices, flash memory devices, or other non-volatile solid state storage devices. Memory 806, optionally, includes one or more storage devices remotely located from CPU(s) 802. Memory 806, or alternately the non-volatile memory device(s) within memory 806, includes a non-transitory computer readable storage medium. In some implementations, memory 806, or the non-transitory computer readable storage medium of memory 806, stores the following programs, modules, and data structures, or a subset or superset thereof: an operating system 816 including procedures for handling various basic system services and for performing hardware dependent tasks; a network communication module 818 for connecting user device 800 to other computers (e.g., a message server system or a terminal device) connected to one or more networks via one or more communication network interfaces 804 (wired or wireless); a presentation module 820 for enabling presentation of information (e.g., a user

interface for a web page or an application program, audio and/or video content, text, etc.) at device 800 via one or more output devices 812 (e.g., displays, speakers, etc.) associated with user interface 810; an input processing module 822 for detecting one or more user inputs or interactions from one of the one or more input devices 814 and interpreting the detected input or interaction; and an information input control module 824 for providing information input control for input boxes in accordance with the embodiments described herein. In some embodiments, the information input control module 824 includes selection detection module 826, regular expression obtaining module 828, data type determination module 830, keyboard selection module 832, and keyboard presentation module 834. In some embodiments, the information input control module 824 also includes character sets 836 for respective stop characters, special and regular validation filter characters, characters with or without backslash conditions, for each data type, and respective associations 828 between available soft keyboards and different data types.

[00103] Each of the above identified elements may be stored in one or more of the previously mentioned memory devices, and corresponds to a set of instructions for performing a function described above. The above identified modules or programs (i.e., sets of instructions) need not be implemented as separate software programs, procedures, modules or data structures, and thus various subsets of these modules may be combined or otherwise re-arranged in various implementations. In some implementations, memory 406, optionally, stores a subset of the modules and data structures identified above. Furthermore, memory 406, optionally, stores additional modules and data structures not described above.

[00104] While particular embodiments are described above, it will be understood it is not intended to limit the invention to these particular embodiments. On the contrary, the invention includes alternatives, modifications and equivalents that are within the spirit and scope of the appended claims. Numerous specific details are set forth in order to provide a thorough understanding of the subject matter presented herein. But it will be apparent to one of ordinary skill in the art that the subject matter may be practiced without these specific details. In other instances, well-known methods, procedures, components, and circuits have not been described in detail so as not to unnecessarily obscure aspects of the embodiments.

[00105] Although some of the various drawings illustrate a number of logical stages in a particular order, stages that are not order dependent may be reordered and other stages may be combined or broken out. While some reordering or other groupings are specifically mentioned, others will be obvious to those of ordinary skill in the art and so do not present an exhaustive list of alternatives. Moreover, it should be recognized that the stages could be implemented in hardware, firmware, software or any combination thereof.

[00106] The foregoing description, for purpose of explanation, has been described with reference to specific embodiments. However, the illustrative discussions above are not intended to be

exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the invention and its practical applications, to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated.

What is claimed is:

Claims

1. A method of input control for text input boxes, comprising:

at a device having one or more processors, memory, and a display:

detecting selection of a text input box in a user interface;

in response to detecting the selection of the text input box and prior to receiving input text for the text input box, obtaining a regular expression for acceptable input corresponding to text input box;

for each character in the regular expression, determining whether the character belongs to a respective validation filter character set for a predetermined data type; and

in response to a determination that all characters in the regular expression belongs to the respective validation filter character set for the predetermined data type, selectively presenting a respective character keyboard corresponding to the predetermined data type to a user for text entry.

2. The method of claim 1, wherein:

the respective validation filter character set includes a subset of regular validation filter characters and a subset of special validation filter characters; and

for each character in the regular expression, determining whether the character belongs to the respective validation filter character set for the predetermined data type further comprises:

determining whether the character belongs to the subset of regular validation filter characters or the subset of special validation filter characters;

in accordance with a determination that the character belongs to the subset of regular validation filter characters, proceeding to process a next character in the regular expression; and

in accordance with a determination that the character belongs to the subset of special validation filter characters, processing all characters that fall between the character and a counterpart character of the character in the regular expression as a respective character subset.

3. The method of claim 2, further comprising:

for each character in the regular expression:

when determining whether the character belongs to the subset of regular validation filter characters or the subset of special validation filter characters, first determining whether the character belongs to a stop character set for the predetermined data type; and

in accordance with a determination that the character does not belong to the stop character set for the predetermined data type, proceeding to determine whether the character belongs to the subset of regular validation filter characters or the subset of special validation filter characters.

4. The method of claim 3, wherein:

the stop character set includes a subset of stop characters having a backslash condition, and a subset of stop characters without any backslash condition; and

determining whether the character belongs to a stop character set for the predetermined data type further comprises:

determining whether the character has an associated backslash;

in accordance with a determination that the character has an associated backslash, determining whether the character belongs to the subset of stop characters having a backslash condition; and

in accordance with a determination that the character does not have any associated backslash, determining whether the character belongs to the subset of stop characters without any backslash condition.

5. The method of claim 1, further comprising:

in response to a determination that a current character in the regular expression does not belong to the respective validation filter character set for the predetermined data type, determining whether each character in the regular expression belongs to a respective validation filter character set for another predetermined data type.

6. A system for input control for text input boxes, comprising:

one or more processors; and

memory having instructions stored thereon, the instructions, when executed by the one or more processors, cause the processors to perform operations comprising:

detecting selection of a text input box in a user interface;

in response to detecting the selection of the text input box and prior to receiving input text for the text input box, obtaining a regular expression for acceptable input corresponding to text input box;

for each character in the regular expression, determining whether the character belongs to a respective validation filter character set for a predetermined data type; and

in response to a determination that all characters in the regular expression belongs to the respective validation filter character set for the predetermined data type, selectively presenting a respective character keyboard corresponding to the predetermined data type to a user for text entry.

7. The system of claim 6, wherein:

the respective validation filter character set includes a subset of regular validation filter characters and a subset of special validation filter characters; and

for each character in the regular expression, determining whether the character belongs to the respective validation filter character set for the predetermined data type further comprises:

determining whether the character belongs to the subset of regular validation filter characters or the subset of special validation filter characters;

in accordance with a determination that the character belongs to the subset of regular validation filter characters, proceeding to process a next character in the regular expression; and

in accordance with a determination that the character belongs to the subset of special validation filter characters, processing all characters that fall between the character and a counterpart character of the character in the regular expression as a respective character subset.

8. The system of claim 7, wherein the operations further comprise:

for each character in the regular expression:

when determining whether the character belongs to the subset of regular validation filter characters or the subset of special validation filter characters, first determining whether the character belongs to a stop character set for the predetermined data type; and

in accordance with a determination that the character does not belong to the stop character set for the predetermined data type, proceeding to determine whether the character belongs to the subset of regular validation filter characters or the subset of special validation filter characters.

9. The system of claim 8, wherein:

the stop character set includes a subset of stop characters having a backslash condition, and a subset of stop characters without any backslash condition; and

determining whether the character belongs to a stop character set for the predetermined data type further comprises:

determining whether the character has an associated backslash;

in accordance with a determination that the character has an associated backslash, determining whether the character belongs to the subset of stop characters having a backslash condition; and

in accordance with a determination that the character does not have any associated backslash, determining whether the character belongs to the subset of stop characters without any backslash condition.

10. The system of claim 6, wherein the operations further comprise:

in response to a determination that a current character in the regular expression does not belong to the respective validation filter character set for the predetermined data type, determining whether each character in the regular expression belongs to a respective validation filter character set for another predetermined data type.

11. A non-transitory computer-readable medium having instructions stored thereon, the instructions, when executed by one or more processors, cause the processors to perform operations comprising:

detecting selection of a text input box in a user interface;

in response to detecting the selection of the text input box and prior to receiving input text for the text input box, obtaining a regular expression for acceptable input corresponding to text input box;

for each character in the regular expression, determining whether the character belongs to a respective validation filter character set for a predetermined data type; and

in response to a determination that all characters in the regular expression belongs to the respective validation filter character set for the predetermined data type, selectively presenting a respective character keyboard corresponding to the predetermined data type to a user for text entry.

12. The computer-readable medium of claim 11, wherein:

the respective validation filter character set includes a subset of regular validation filter characters and a subset of special validation filter characters; and

for each character in the regular expression, determining whether the character belongs to the respective validation filter character set for the predetermined data type further comprises:

determining whether the character belongs to the subset of regular validation filter characters or the subset of special validation filter characters;

in accordance with a determination that the character belongs to the subset of regular validation filter characters, proceeding to process a next character in the regular expression; and

in accordance with a determination that the character belongs to the subset of special validation filter characters, processing all characters that fall between the character and a counterpart character of the character in the regular expression as a respective character subset.

13. The computer-readable medium of claim 12, wherein the operations further comprise:

for each character in the regular expression:

when determining whether the character belongs to the subset of regular validation filter characters or the subset of special validation filter characters, first determining whether the character belongs to a stop character set for the predetermined data type; and

in accordance with a determination that the character does not belong to the stop character set for the predetermined data type, proceeding to determine whether the character belongs to the subset of regular validation filter characters or the subset of special validation filter characters.

14. The computer-readable medium of claim 13, wherein:

the stop character set includes a subset of stop characters having a backslash condition, and a subset of stop characters without any backslash condition; and

determining whether the character belongs to a stop character set for the predetermined data type further comprises:

determining whether the character has an associated backslash;

in accordance with a determination that the character has an associated backslash, determining whether the character belongs to the subset of stop characters having a backslash condition; and

in accordance with a determination that the character does not have any associated backslash, determining whether the character belongs to the subset of stop characters without any backslash condition.

15. The computer-readable medium of claim 14, wherein the operations further comprise:

in response to a determination that a current character in the regular expression does not belong to the respective validation filter character set for the predetermined data type, determining whether each character in the regular expression belongs to a respective validation filter character set for another predetermined data type.

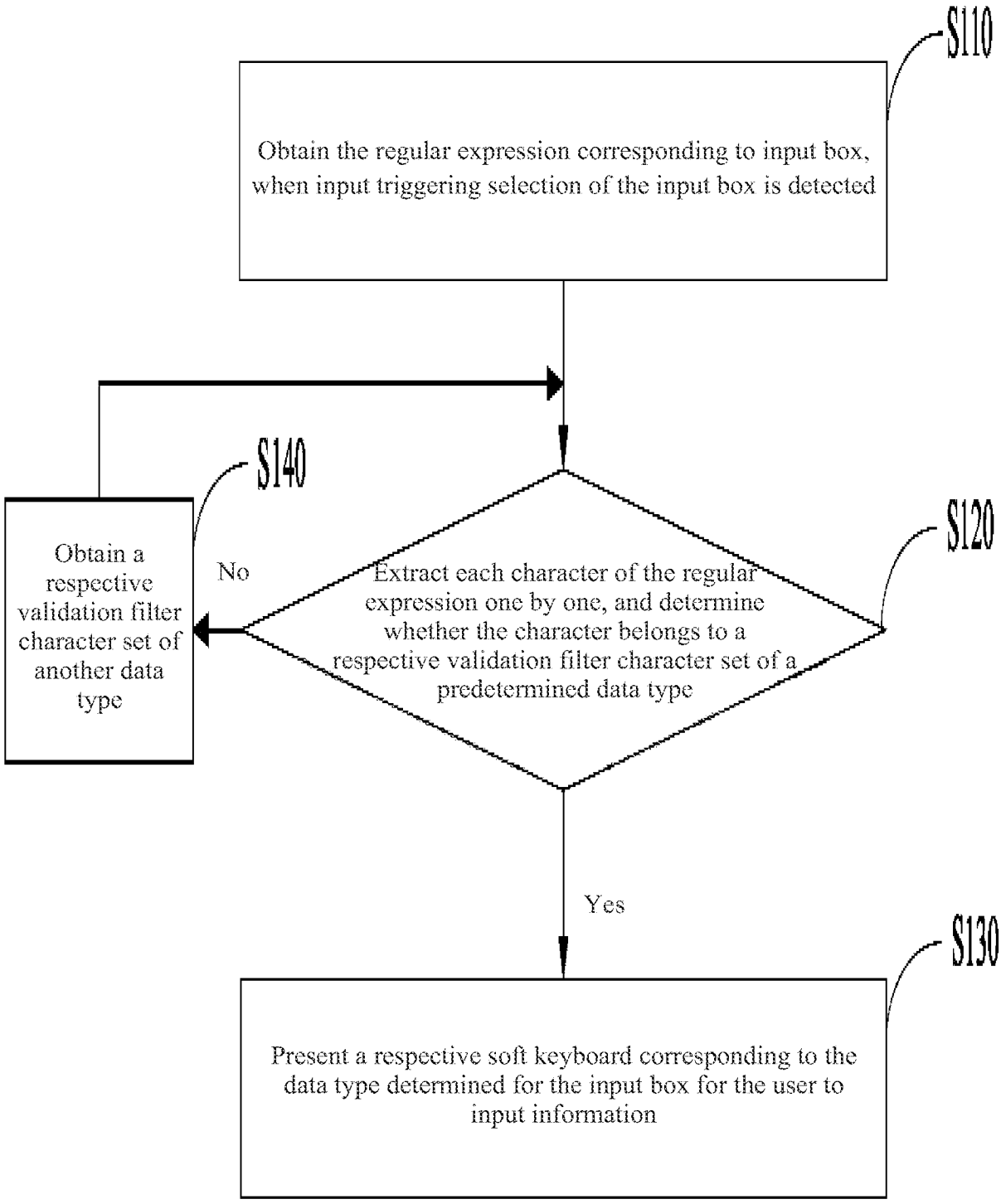

S110
Obtain the regular expression corresponding to input box, when input triggering selection of the input box is detected
S120
Extract each character of the regular expression one by one, and determine whether the character belongs to a respective validation filter character set of a predetermined data type
No
S140
Obtain a respective validation filter character set of another data type
Yes
S130
Present a respective soft keyboard corresponding to the data type determined for the input box for the user to input information

Fig. 1

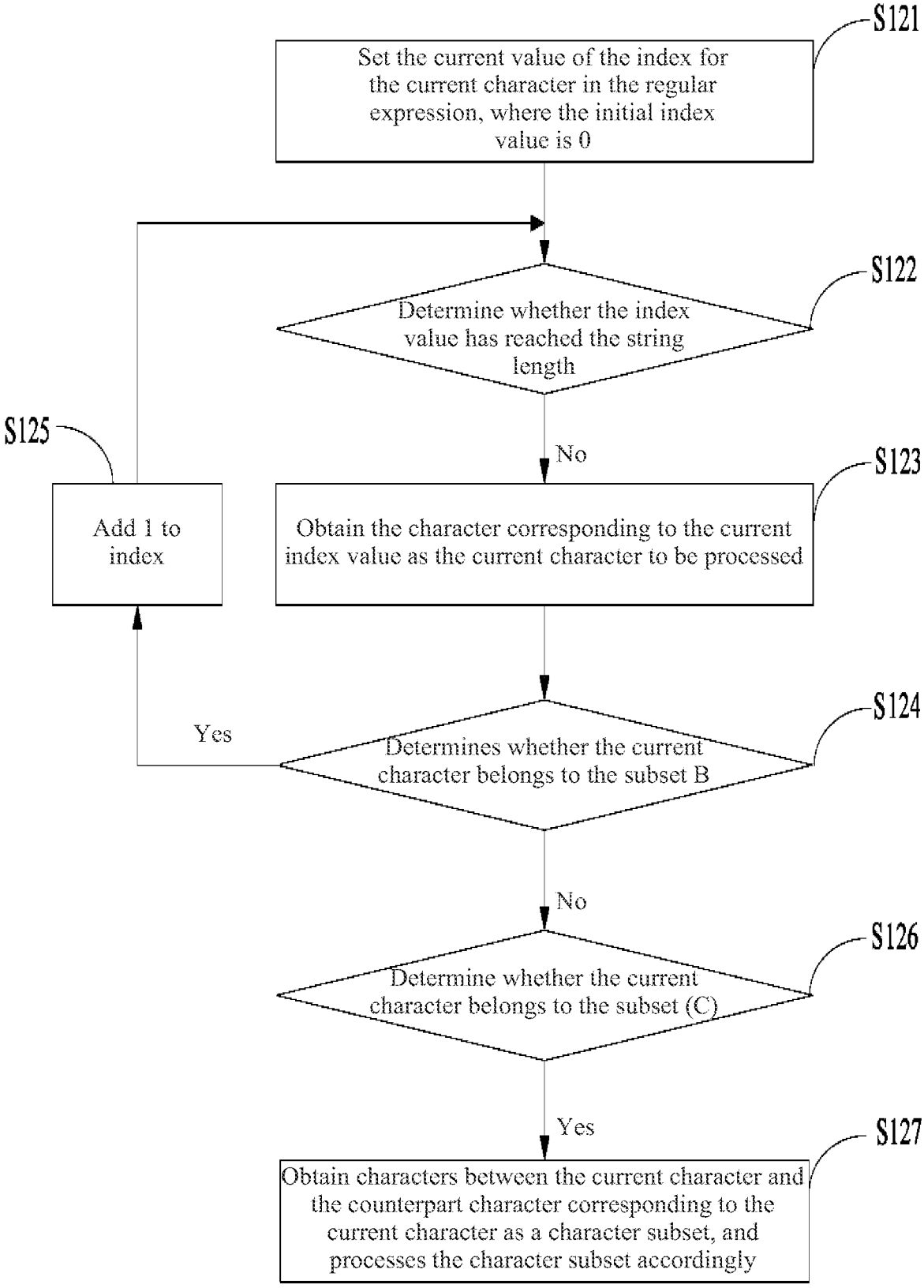

S121
Set the current value of the index for the current character in the regular expression, where the initial index value is 0
S122
Determine whether the index value has reached the string length
No
S123
Obtain the character corresponding to the current index value as the current character to be processed
S124
Determines whether the current character belongs to the subset B
Yes
S125
Add 1 to index
No
S126
Determine whether the current character belongs to the subset (C)
Yes
S127
Obtain characters between the current character and the counterpart character corresponding to the current character as a character subset, and processes the character subset accordingly

Fig. 2

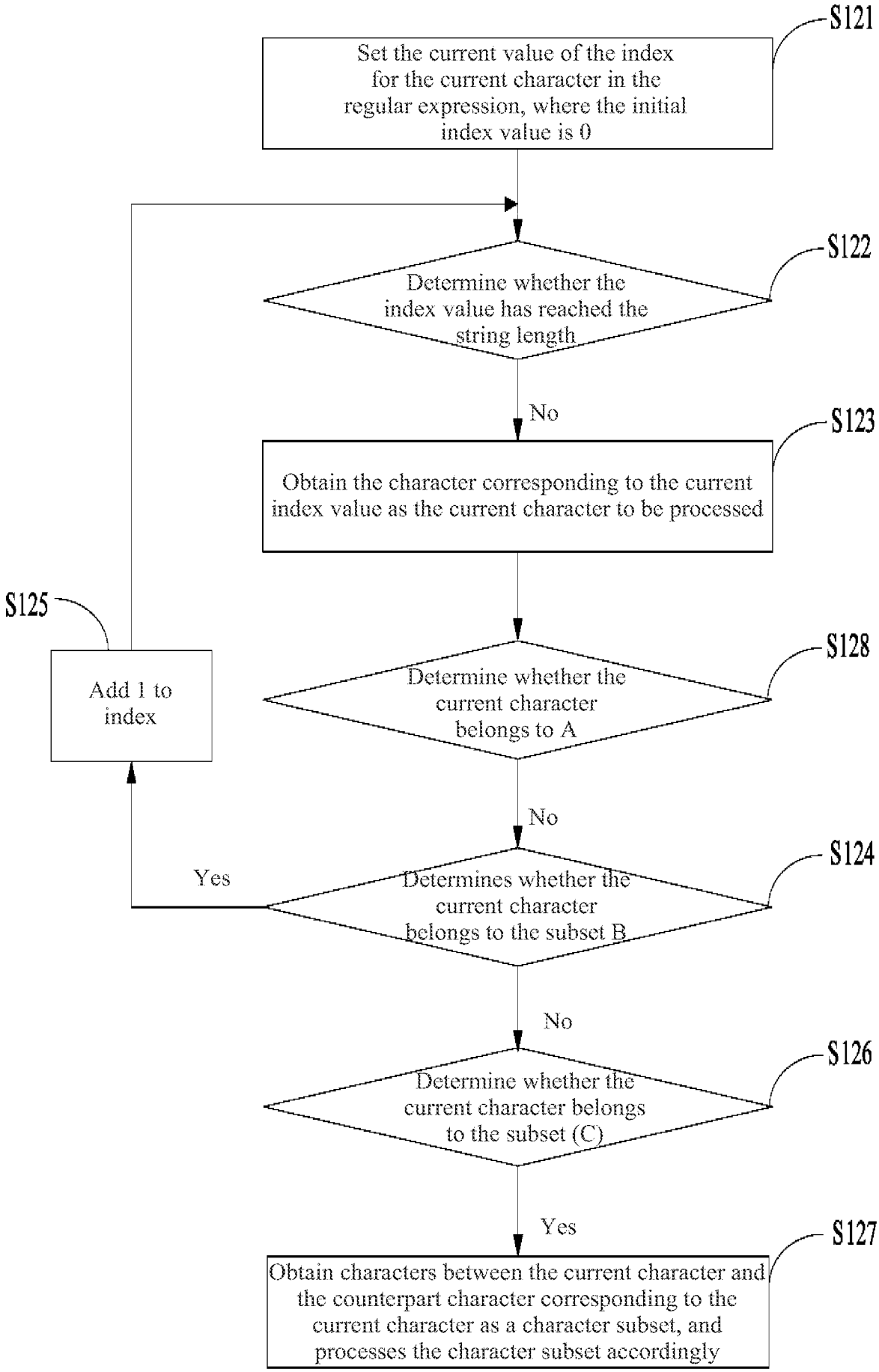

S121
Set the current value of the index for the current character in the regular expression, where the initial index value is 0
S122
Determine whether the index value has reached the string length
No
S123
Obtain the character corresponding to the current index value as the current character to be processed
S125
Add 1 to index
S128
Determine whether the current character belongs to A
No
Yes
S124
Determines whether the current character belongs to the subset B
No
S126
Determine whether the current character belongs to the subset (C)
Yes
S127
Obtain characters between the current character and the counterpart character corresponding to the current character as a character subset, and processes the character subset accordingly

Fig. 3

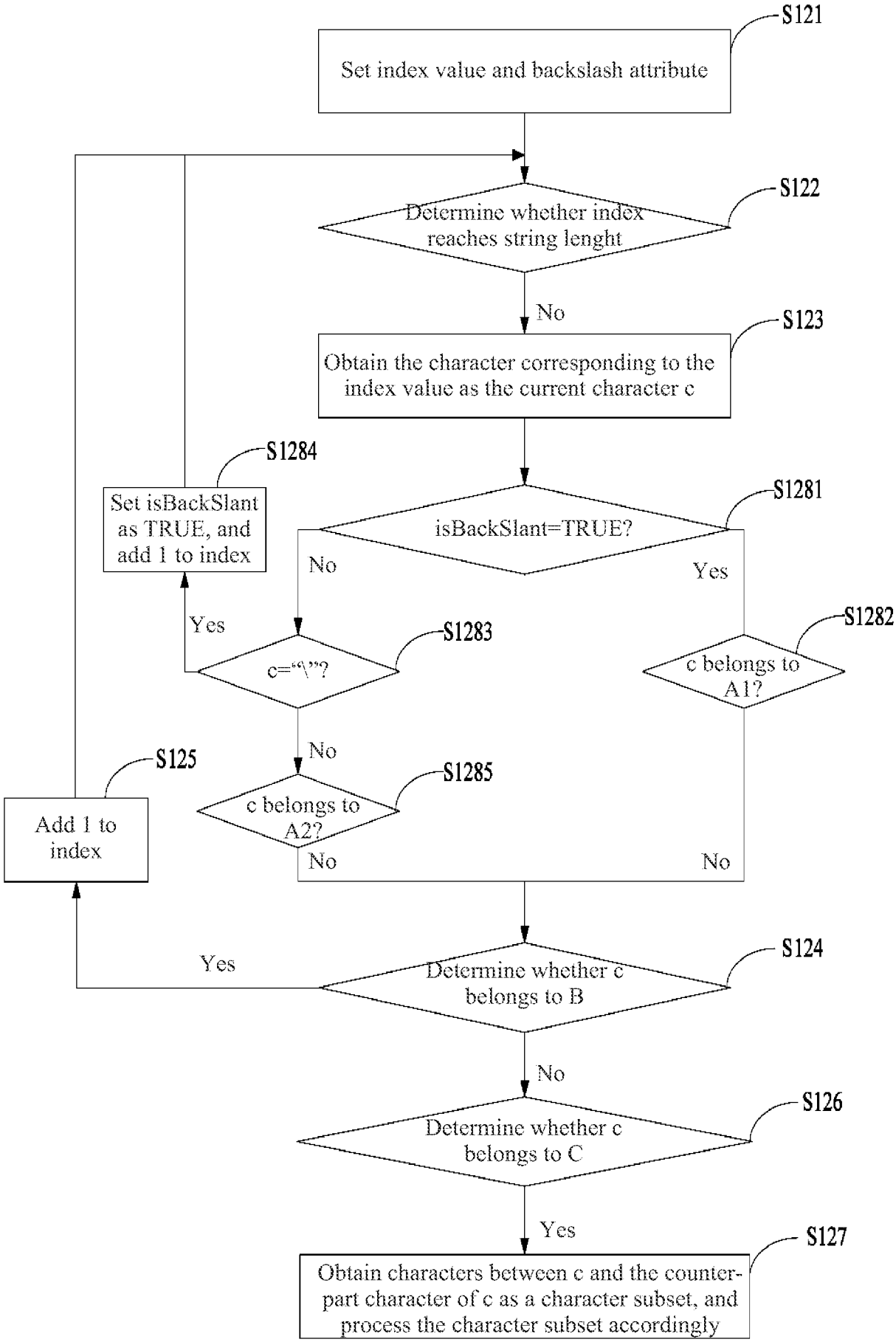

S121
Set index value and backslash attribute
S122
Determine whether index reaches string lenght
No
S123
Obtain the character corresponding to the index value as the current character c
S1284
Set isBackSlant as TRUE, and add 1 to index
S1281
isBackSlant=TRUE?
No
Yes
Yes
S1283
c="\"?
S1282
c belongs to A1?
No
S125
S1285
c belongs to A2?
Add 1 to index
No
No
Yes
S124
Determine whether c belongs to B
No
S126
Determine whether c belongs to C
Yes
S127
Obtain characters between c and the counter-part character of c as a character subset, and process the character subset accordingly

Fig. 4

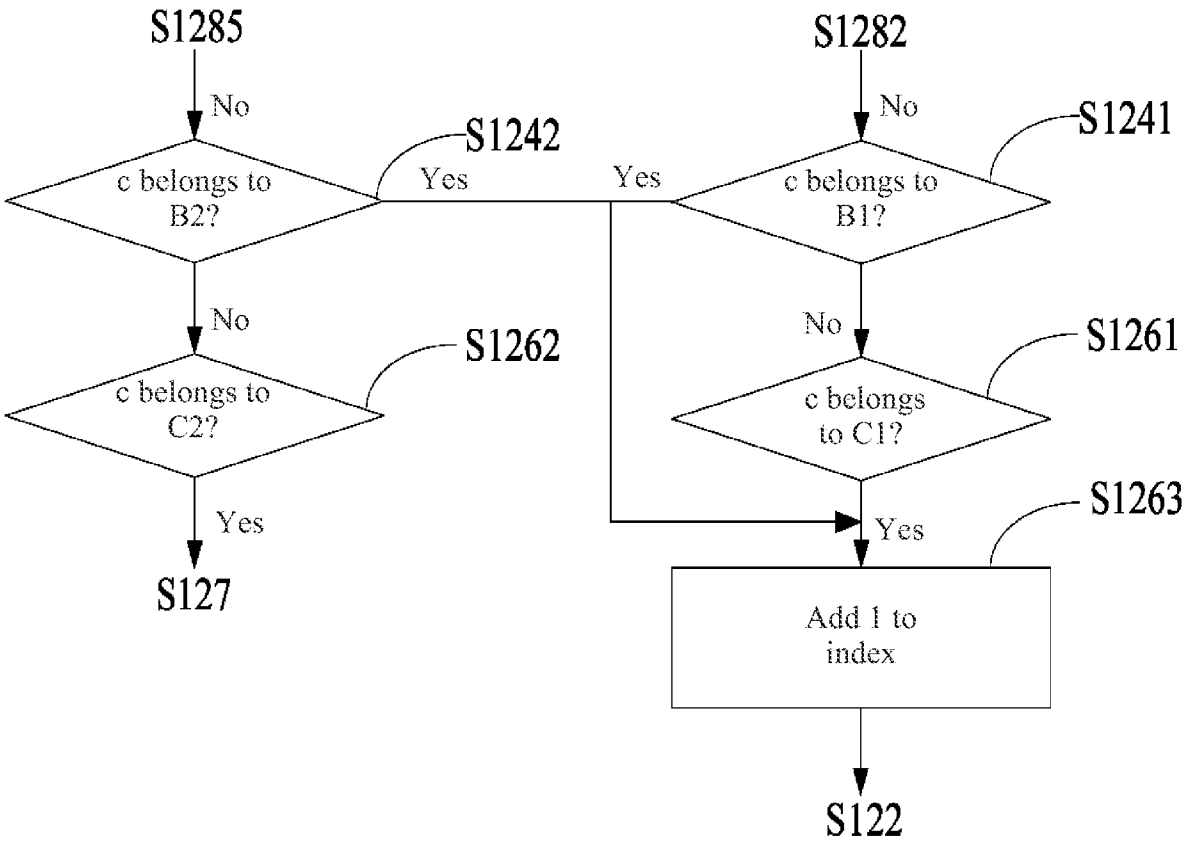

S1285
No
c belongs to B2?
S1242
Yes
S1282
No
c belongs to B1?
S1241
Yes
No
c belongs to C2?
S1262
Yes
S127
No
c belongs to C1?
S1261
Yes
S1263
Add 1 to index
S122

Fig. 5

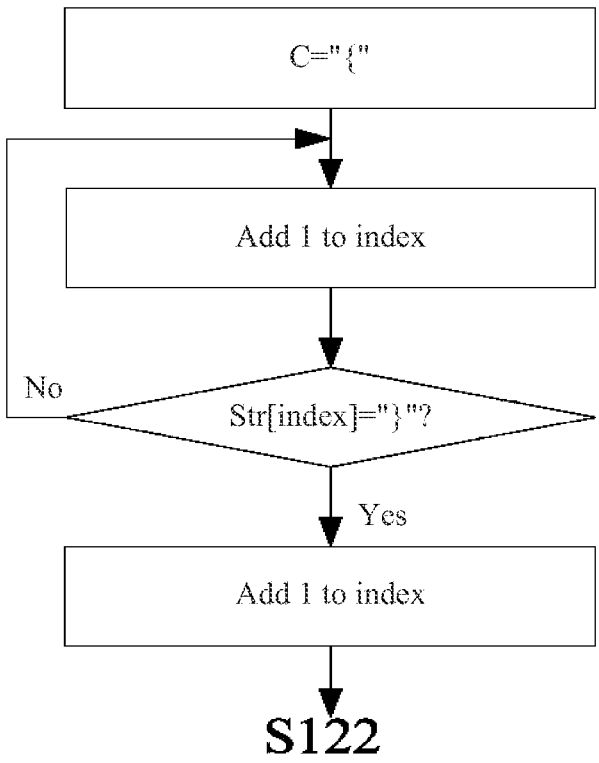

C="{"
Add 1 to index
No
Str[index]="}"?
Yes
Add 1 to index
S122

Fig. 6a

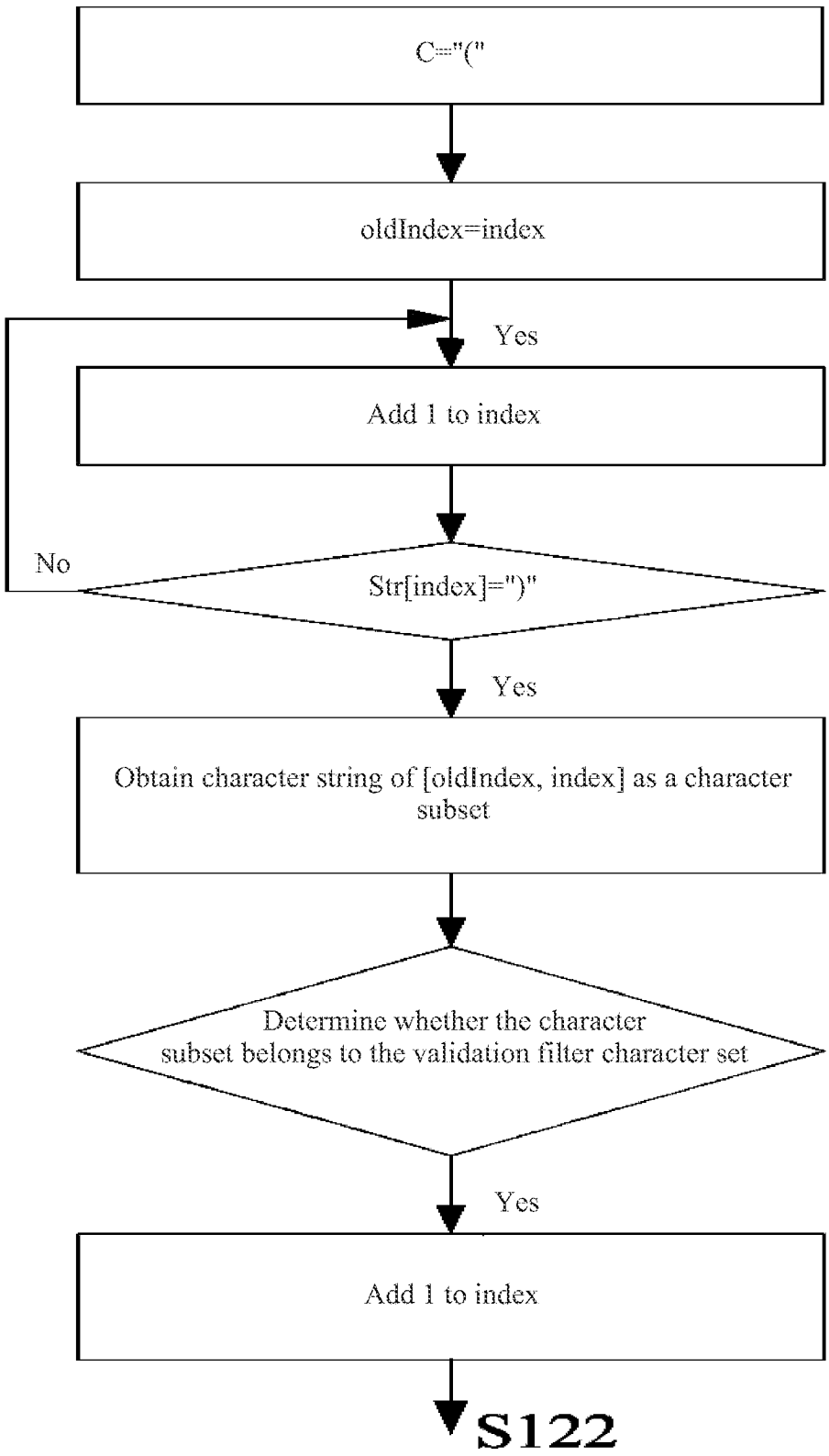

C=="("
oldIndex=index
Yes
Add 1 to index
No
Str[index]=")"
Yes
Obtain character string of [oldIndex, index] as a character subset
Determine whether the character subset belongs to the validation filter character set
Yes
Add 1 to index
S122

Fig. 6b

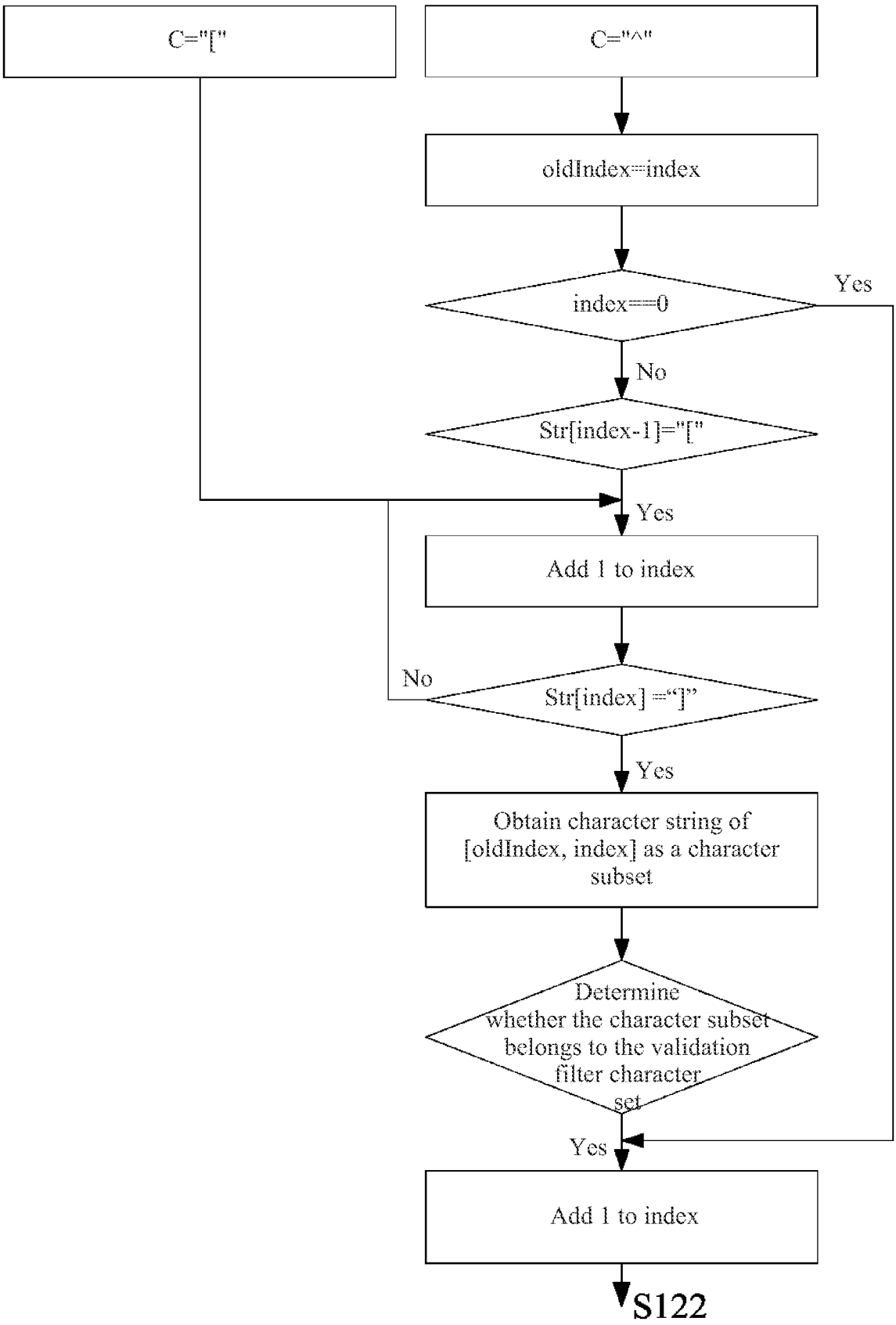

C="["
C="^"
oldIndex=index
index==0
Yes
No
Str[index-1]="["
Yes
Add 1 to index
No
Str[index] ="]"
Yes
Obtain character string of [oldIndex, index] as a character subset
Determine whether the character subset belongs to the validation filter character set
Yes
Add 1 to index
S122

Fig. 6c

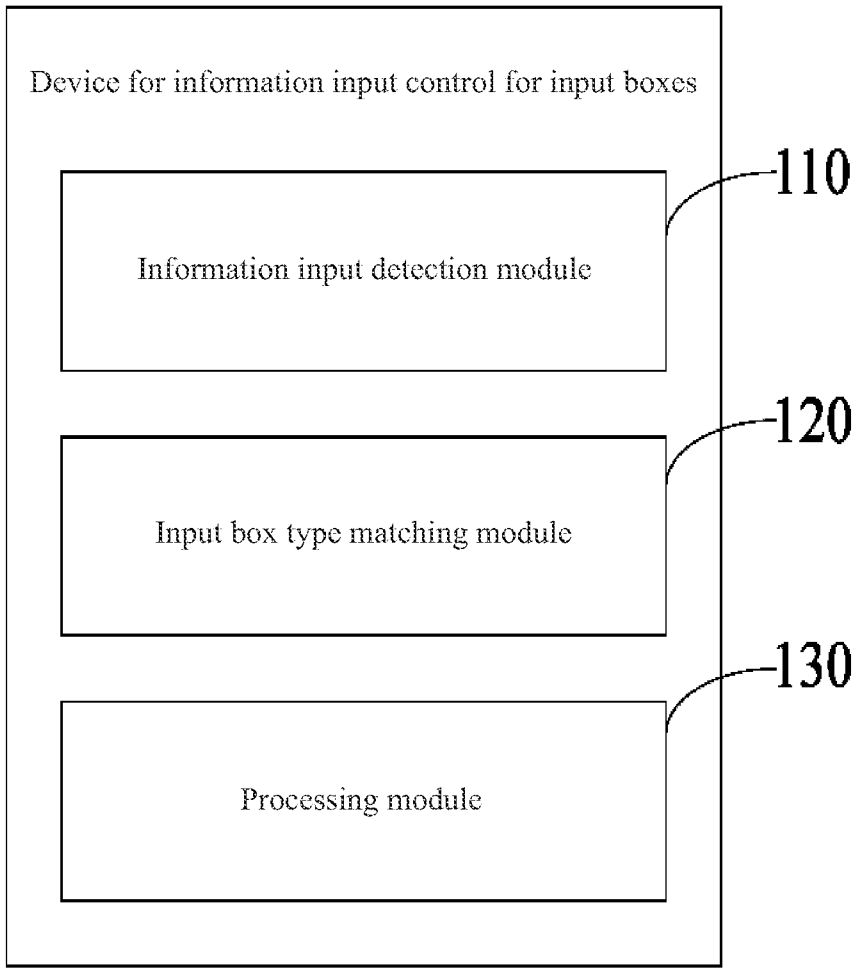

Device for information input control for input boxes
Information input detection module
110
Input box type matching module
120
Processing module
130

Fig. 7

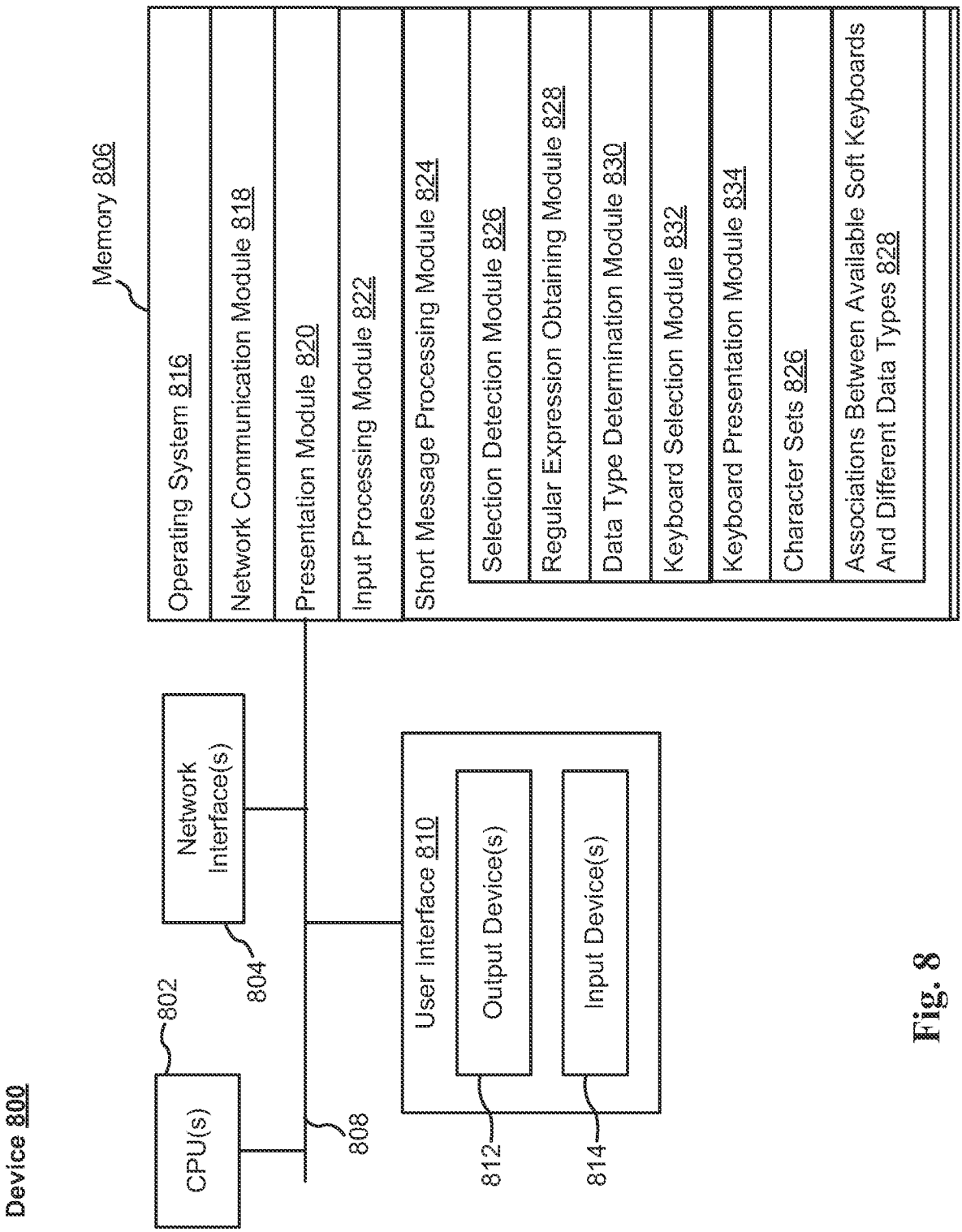

Device 800
CPU(s)
802
Network Interface(s)
804
808
User Interface 810
Output Device(s)
Input Device(s)
812
814
Memory 806
Operating System 816
Network Communication Module 818
Presentation Module 820
Input Processing Module 822
Short Message Processing Module 824
Selection Detection Module 826
Regular Expression Obtaining Module 828
Data Type Determination Module 830
Keyboard Selection Module 832
Keyboard Presentation Module 834
Character Sets 826
Associations Between Available Soft Keyboards And Different Data Types 828

Fig. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2014/070098

A. CLASSIFICATION OF SUBJECT MATTER

G06F 3/0481(2013.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 3/-

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)

CNTXT,WPI, EPODOC: input+, box?, frame?, field?, virtual, character, keyboard, regular, expression?, regex, regexp, data, type, Chinese, English, digit+, numeric, number

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CN 101916158 A (BAIDU ON-LINE NETWORK TECHNOLOGY CO., LTD.) 15 December 2010 (2010-12-15) claims 1-22, description paragraphs [0020]-[0040]	1, 5, 6, 10, 11
Y	US 2009132950 A1 (INT BUSINESS MACHINES CORP) 21 May 2009 (2009-05-21) claims 1-3, figure 1	1, 5, 6, 10, 11
A	CN 101916158 A (BAIDU ON-LINE NETWORK TECHNOLOGY CO., LTD.) 15 December 2010 (2010-12-15) the whole document	2-4, 7-9, 12-15
A	US 2009132950 A1 (INT BUSINESS MACHINES CORP) 21 May 2009 (2009-05-21) the whole document	2-4, 7-9, 12-15

☐ 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
31 March 2014	**22 April 2014**

Name and mailing address of the ISA/	Authorized officer
STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA(ISA/CN) **6,Xitucheng Rd., Jimen Bridge, Haidian District, Beijing, China** **100088 China**	**SONG,Yunyun**
Facsimile No. **(86-10)62019451**	Telephone No. **(86-10)62089120**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2014/070098

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 101916158A	15 December 2010	None	
US 2009132950A1	21 May 2009	US 7885916B2	08 February 2011