

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
23 October 2008 (23.10.2008)

PCT

(10) International Publication Number
WO 2008/127939 A1

- (51) International Patent Classification:
G06F 17/28 (2006.01) **G06F 17/00** (2006.01)
- (21) International Application Number:
PCT/US2008/059760
- (22) International Filing Date: 9 April 2008 (09.04.2008)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
60/911,279 11 April 2007 (11.04.2007) US
200710101819.7 16 April 2007 (16.04.2007) CN
- (71) Applicant (for all designated States except US):
GOOGLE INC. [US/US]; 1600 Amphitheatre Park-
way, Mountain View, California 94043 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **HONG, Feng**
[CN/US]; 807 Ram Lane, Foster City, California 94404
(US). **ZHAO, Yuke** [US/US]; P. O. Box 56, Mountain
View, California 94042 (US). **GUO, Quji** [CN/US]; 1228
Coronado Drive, #17, Sunnyvale, California 94086 (US).
- (74) Agent: **FRANZ, Paul E.**; Fish & Richardson P.c., P. O.
Box 1022, Minneapolis, Minnesota 55440-1022 (US).

(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, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declaration under Rule 4.17:

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

Published:

— with international search report

[Continued on next page]

(54) Title: INPUT METHOD EDITOR INTEGRATION

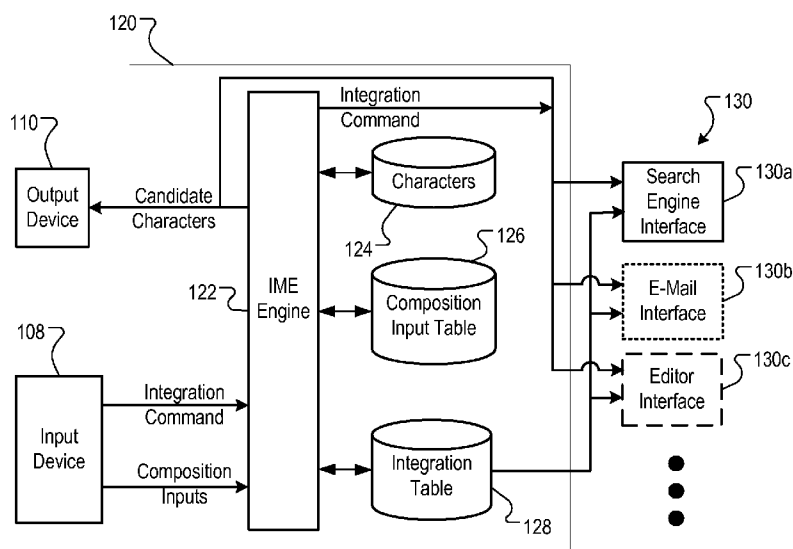


FIG. 2

(57) Abstract: Methods, systems, and apparatus, including computer program products, in which an input method editor receives composition inputs in an input method editor environment, the composition inputs based on user inputs. Candidate selections are identified in the input method editor environment based on the composition inputs. Each candidate selection comprises one or more ideograms. The input method editor environment receives a search command based on a user input, and selects one of the candidate selections in response to the search command. The selected candidate selection and the search command are provided to a search engine interface external to the input method editor environment.

WO 2008/127939 A1



-
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

INPUT METHOD EDITOR INTEGRATION

This application claims the benefit of priority of U.S. Provisional Patent application no. 60/911,279, filed on April 11, 2007, and China application no. 200710101819.7. filed April 16, 2007, the entire disclosures of which are incorporated herein by reference.

5

BACKGROUND

This disclosure relates to input methods.

10

Languages that use a logographic script in which one or two characters correspond roughly to one word or meaning have more characters than keys on a standard input device, such as a computer keyboard or a mobile device keypad. For example, the Chinese language contains tens of thousands of characters having pronunciations defined by base Pinyin elements and five tones. The mapping of these potentially many-to-one associations can be implemented by input methods that facilitate entry of characters and symbols not found on input devices. Accordingly, a Western style keyboard can be used to input Chinese characters. Likewise, input methods can be used for using a Western style keyboard or some other input device to input many other languages that include ideograms, such as the Japanese language, the Korean language, and other languages.

15

An input method editor user interface may be associated with an input method editor. Often a user may desire to utilize the input method editor to generate characters, e.g., Hanzi characters, and input the characters into another application, e.g., a search engine input field. For example, a user may open a browser and access a search engine interface, and then open an input method editor and enter a Chinese query in the input method editor environment, e.g., the input method editor user interface. The user may have to select a particular set of characters in the input method editor environment and then activate a search command in the search engine interface.

20

The repetition of the manual selections and/or switching operations can be detrimental to a user experience. Eliminating one or more of these manual selections and/or switching operations can thus improve the user experience.

25

SUMMARY

30

This specification describes technologies related to input method editors. In some implementations, an input method editor receives composition inputs in an input method editor environment. The composition inputs are based on user inputs. Candidate selections

are identified in the input method editor environment based on the composition inputs. Each candidate selection comprises one or more ideograms. The input method editor environment receives a search command based on a user input, and selects one of the candidate selections in response to the search command. The selected candidate selection and the search
5 command are provided to a search engine interface external to the input method editor environment.

In some implementations, an input method editor data store stores integration commands, each integration command identifying an integration of the input method editor with a corresponding application environment that is external to an input method editor user
10 interface. Candidate selections based on one or more composition inputs can be identified in the input method editor user interface. The input method editor user interface receives an integration command based on a user input, and one of the candidate selections is selected in response to the integration command. The selected candidate selection is provided to the corresponding application environment external to the input method editor user interface.

In some implementations, a system includes a data store storing one or more input
15 method editor data store integration commands. Each integration command identifies an integration of an input method editor with a corresponding application instance that is external to an input method editor user interface. The system also includes a plurality of application interfaces. Each application interface is associated with one of the corresponding
20 application instances and configured to facilitate data communication from an input method editor to the application instance. The system also includes an input method editor configured to identify candidate selections based on one or more composition inputs and, in response to receiving an integration command associated with an application interface, access the application interface, select one of the candidate selections and provide the selected
25 candidate selection to the corresponding application instance associated with the application interface.

The details of one or more embodiments of the subject matter described in this specification are set forth in the accompanying drawings and the description below. Other
30 features, aspects, and advantages of the subject matter will become apparent from the description, the drawings, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of an example device that can be utilized to implement the systems and methods described herein.

Fig. 2 is a block diagram of an example input method editor system.

5 Figs. 3 - 5 are screen shots of an example input method editor environment.

Fig. 6 is a flow diagram of an example search integration process.

Fig. 7 is a flow diagram of a process for selecting a candidate selection.

Like reference numbers and designations in the various drawings indicate like elements.

10

DETAILED DESCRIPTION

Fig. 1 is a block diagram of an example device 100 that can be utilized to implement the systems and methods described herein. The example device 100 can, for example, be a computer device, such as a personal computer device, or other electronic devices, such as a mobile phone, mobile communication device, personal digital assistant (PDA), and the like.

15 The example device 100 includes a processing device 102, a first data store 104, a second data store 106, input devices 108, output devices 110, and a network interface 112. A bus system 114, including, for example, a data bus and a motherboard, can be used to establish and control data communication between the components 102, 104, 106, 108, 110 and 112. Other system architectures can also be used.

20 The processing device 102 can, for example, include one or more microprocessors. The first data store 104 can, for example, include a random access memory storage device, such as a dynamic random access memory, or other types of computer-readable medium memory devices. The second data store 106 can, for example, include one or more hard drives, a flash memory, and/or a read only memory, or other types of computer-readable
25 medium memory devices.

Example input devices 108 can include a keyboard, a mouse, a stylus, etc., and example output devices 110 can include a display device, an audio device, etc. The network interface 112 can, for example, include a wired or wireless network device operable to communicate data to and from a network 116. The network 116 can include one or more
30 local area networks (LANs) and/or a wide area network (WAN), such as the Internet.

In some implementations, the device 100 can include input method editor (IME) code 101 in a data store, such as the data store 106. The input method editor code 101 can include instructions that upon execution cause the processing device 102 to carry out input method

editing functions. The input method editor code 101 can, for example, comprise interpreted instructions, such as script instructions, e.g., JavaScript or ECMAScript instructions, that can be executed in a web browser environment. Other implementations can also be used, e.g., a stand-alone application, an applet, a plug-in module, etc.

5 Execution of the input method editor code 101 generates or launches an input method editor instance (IMEI) 103. The input method editor instance 103 can define an input method editor environment, e.g., user interface, and can facilitate the processing of one or more input methods at the device 100, during which time the device 100 can receive composition inputs for input characters, ideograms, or symbols, such as, for example, Hanzi characters. For
10 example, the user can use one or more of the input devices 108 (e.g., a keyboard, such as a Western-style keyboard, a stylus used with a handwriting recognition engine, etc.) to input composition inputs for identification of Hanzi characters. In some examples, a Hanzi character can be composed of more than one composition input.

 The first data store 104 and/or the second data store 106 can store an association of
15 composition inputs and characters. Based on a user input, the input method editor instance 103 can use information in the data store 104 and/or the data store 106 to identify one or more candidate selections represented by the input. In some implementations, if more than one candidate selection is identified, the candidate selections are displayed on an output device 110. Using the input device 108, the user can select from the candidate selections a
20 Hanzi character that the user desires to input.

 In some implementations, the input method editor instance 103 on the device 100 can receive one or more Pinyin composition inputs and convert the composition inputs into Hanzi characters. The input method editor instance 103 can, for example, use compositions of Pinyin syllables or characters received from keystrokes to represent the Hanzi characters.
25 Each Pinyin syllable can, for example, correspond to one or more keystrokes in the Western-style keyboard. Using a Pinyin input method editor, a user can input a Hanzi character by using composition inputs that includes one or more Pinyin syllables representing the sound of the Hanzi character. Input methods for other languages, however, can also be facilitated.

30 Other application software 105 can also be stored in data stores 104 and/or 106, including web browsers, word processing programs, e-mail clients, etc. Each of these applications can generate a corresponding application instance (AI) 107. Each application instance can define an environment that can facilitate a user experience by presenting data to the user and facilitating data input from the user. For example, web browser software can

generate a search engine environment; e-mail software can generate an e-mail environment; a word processing program can generate an editor environment; etc.

In some implementations, a remote computing system 118 having access to the system 100 can also be used to edit a logographic script. For example, the system 100 may be a server that provides logographic script editing capability via the network 116. In one example, a user can edit a logographic script stored in the data store 104 and/or the data store 106 using a remote computing system, e.g., a client computer. The system 100 can, for example, select a character and receive a composition input from a user over the network interface 112. The processing device 102 can, for example, identify one or more characters adjacent to the selected character, and identify one or more candidate selections based on the received composition input and the adjacent characters. The system 100 can transmit a data communication that includes the candidate selections back to the remote computing system.

Fig. 2 is a block diagram of an example input method editor system 120. The input method editor system 120 can, for example, be implemented using the input method editor code 101 and associated data stores 104 and 106. The input method editor system 120 includes an input method editor engine 122, a character data store 124, a composition input data store 126, and an integration table 128. Other storage architectures can also be used. The character data store 124 includes characters of a logographic script used in one or more language models, and characters and words in Roman-based or western-style alphabets, e.g., English, German, Spanish, etc. In some implementations, the character data store 124 also includes information about relationships between characters. For example, the character data store 124 can include scores or probability values assigned to a character depending on characters adjacent to the character. Other contextual relation data can also be used.

The composition input data store 126 includes an association of composition inputs and the characters stored in the character data store 124. In some implementations, the composition input data store 126 can link each of the characters in the character data store 124 to a composition input used by the input method editor engine 122. For example, the input method editor engine 122 can use the information in the character data store 124 and the composition input data store 126 to associate and/or identify one or more characters in the character data store 124 with one or more composition inputs in the composition input data store 126.

The integration table data store 128 can store integration command data related to integration commands. Each integration command can, for example, identify an integration of the input method editor 120 with a corresponding application interface 130. The

application interface 130 can facilitate communication with a corresponding application or application environment that is external to the input method editor instance 103 environment.

In some implementations, an application interface 130 can include instructions, e.g., a script or compiled code, and can be invoked based on the integration command received. In some implementations, the application interfaces 130 can be stored in a data store associated with an application, and the integration table 128 can include data identifying and providing access to the corresponding application interface 130. For example, a search engine interface 130a can be a script stored in a browser cache; an e-mail interface 130b can be a script stored in an e-mail cache; and an editor interface 130c can be compiled code stored in an editor data store.

Other implementations can also be used. For example, the application interfaces 130 can be application program interfaces to corresponding applications, or can be an integration with an application user interface, for example.

The input method editor instance 103 can, for example, receive in the input method editor user interface an integration command. The integration command can be based on a user input and received from a user device 108. In response to the integration command, one of the candidate selections can be selected and provided to an application interface 130 with the integration command. The application interface 130 can facilitate communication with an application or application environment that is external to the input method editor instance 103 environment.

For example, if the integration command is a search command, and the integration table 128 defines the search command to be associated with the search engine interface 130a, then one of the candidate selections can be selected in response to the search command and provided with the search command to a search engine interface 130a. The search engine interface 130a can, for example, provide the candidate selection as input to a search engine and the search command can invoke a search function that utilizes the candidate selection as query data. In some implementations, the integration command is invoked in the input method editor interface, e.g., by a corresponding integration command activation element, e.g. a button, or key stroke.

In some implementations, the input method editor 120, in response to the search command, can determine if a corresponding application interface 130 is active. For example, in response to a search command, input method editor 120 can determine if the search engine interface 130a, e.g., a search engine environment, or a search engine program interface, is active. If the search engine interface 130 is determined to be active, the selected candidate

selection can be provided to the search engine interface 130a as a search input, and a search can be invoked through the search engine interface 130a that causes a search engine to perform a search based on the selected candidate selection.

If the search engine interface 130 is determined to not be active, the input method editor 120 can, for example, instantiate the search engine interface to provide the selected candidate selection and search command.

In some implementations, the candidate selections in the input method editor 120 can be ranked and presented in the input method editor according to the rank. For example, Fig. 3 shows an example input method editor environment 300 presenting five ranked candidate selections 302. The candidate selections 302 are identified based on the Pinyin inputs 304. An integration command button 306, e.g., a search command button 306, is also displayed in the input method editor environment. A selection indicator 308 surrounds the first candidate selection, i.e., 谷歌, indicating that the first candidate selection will be selected in response to an activation of the search command button 306. In some implementations, first candidate selection can be selected by default upon activation of the search command button.

As shown in Fig. 4, the search command button 306 has been activated, and thus the first candidate selection 谷歌 has been provided to a search engine as an input query for a search function.

In some implementations, a user can manually select a candidate selection. For example, a user may desire to invoke a search on a query defined by the second candidate selection. As shown in Fig. 5, the user can, for example, utilize up and down arrow keys to move the selection indicator 308 to select the second candidate selection. Accordingly, activation of the search command button 306 will provide the second candidate selection and a search command to a search engine.

In some implementations, the integration command can be invoked by a keyboard input. For example, a Ctrl-G input can be used to input an integration command.

Although only one integration command button 306 is shown, additional integration command buttons can also be included in the input method environment, e.g., an integration command button for an e-mail program could also be included in the input method environment. Accordingly, multiple integration command buttons can be displayed at the same time to facilitate processing of different integration commands from a single environment.

In some implementations, one or more integration commands that are made available can be determined by an identified context during runtime, e.g., a set Chinese candidate selections during runtime may be examined to determine which integration commands are instantiated. For example, if a user types an input for a city name, an integration command to obtain current weather data of the city may appear; if user types an input for a stock name, an integration command to obtain a stock quote for the stock name may appear, etc. Accordingly, a context defined by candidate selections can be identified, and integration commands related to the identified context can also be identified. The integration commands can thus be enabled based on the context during runtime.

While an example integration with a browser and search engine environment has been described, other integrations can also be realized. For example, any external application that can process an input method input or an ideogram input can be integrated with an input method editor, e.g., an e-mail application, a programming application, etc.

Fig. 6 is a flow diagram of an example search integration process 600. The process 600 can, for example, be implemented in the device 100 of Fig. 1 and/or in the input method editor system 120 of Fig. 2.

The process 600 receives composition inputs in an input method editor environment (602). For example, the device 100 of Fig. 1 implementing an input method editor and/or the input method editor system 120 of Fig. 2 can receive composition inputs based on user inputs from a keyboard device.

The process 600 identifies candidate selections in the input method editor environment based on the composition inputs (604). For example, the device 100 of Fig. 1 implementing an input method editor and/or the input method editor system 120 of Fig. 2 can identify candidate selections, such as the candidate selections shown in Figs. 3 – 5.

The process 600 receives in the input method editor environment a search command (606). For example, the device 100 of Fig. 1 implementing an input method editor and/or the input method editor system 120 of Fig. 2 can receive a search command. The search command can, for example, be generated by a command button in an input method editor environment, or can be generated by a keyboard input when the input method editor environment is active.

The process 600 selects one of the candidate selections in response to the search command (608). For example, the device 100 of Fig. 1 implementing an input method editor and/or the input method editor system 120 of Fig. 2 can perform a default selection of a first

ranked candidate selection upon receiving a search command, or can select a user selected candidate selection upon receiving a search command.

The process 600 provides the selected candidate selection and the search command to a search engine interface (610). For example, the device 100 of Fig. 1 implementing an input method editor and/or the input method editor system 120 of Fig. 2 can provide the candidate selection and the search command to a search engine interface 130a, e.g. a search engine application program interface, or a script that is executed and that launch a web browser resolved to a search engine interface page, etc. The search engine interface can facilitate communication with a corresponding search engine application or application environment that is external to the input method editor environment.

Fig. 7 is a flow diagram of a process 700 for selecting a candidate selection. The process 700 can, for example, be implemented in the device 100 of Fig. 1 and/or in the input method editor system 120 of Fig. 2.

The process 700 ranks the candidate selections (702). For example, the device 100 of Fig. 1 implementing an input method editor and/or the input method editor system 120 of Fig. 2 can rank the candidate selections in response to composition inputs. The ranking can, for example, be based on a likelihood that the candidate selections correspond to the composition inputs.

The process 700 presents the candidate selections in the input method editor environment according to the rank (704). For example, the device 100 of Fig. 1 implementing an input method editor and/or the input method editor system 120 of Fig. 2 can present the candidate selections in an input method editor environment as shown in Figs. 3 – 5.

The process 700 automatically selects a first ranked candidate selection (706). For example, the device 100 of Fig. 1 implementing an input method editor and/or the input method editor system 120 of Fig. 2 can generate a default selection of the first ranked candidate selection, as shown in Figs. 3 and 4. Upon receiving an integration command, the first candidate selection can be selected.

Although described in the context of particular language and input method examples, the systems and methods described herein can be used for any language that utilizes a logographic script. For example, the systems and methods described herein can be used for many other languages that include ideograms, such as the Japanese language, the Korean language, and other languages.

Embodiments of the subject matter and the functional operations described in this specification can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Embodiments of the subject matter described in this specification can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a tangible program carrier for execution by, or to control the operation of, data processing apparatus. The tangible program carrier can be a propagated signal or a computer readable medium. The propagated signal is an artificially generated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus for execution by a computer. The computer readable medium can be a machine-readable storage device, a machine-readable storage substrate, a memory device, a composition of matter effecting a machine-readable propagated signal, or a combination of one or more of them.

The term “data processing apparatus” encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them.

A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, or declarative or procedural languages, and it can be deployed in any form, including as a stand alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

The processes and logic flows described in this specification can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows

can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit).

Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Moreover, a computer can be embedded in another device, e.g., a mobile telephone, a personal digital assistant (PDA), a mobile audio or video player, a game console, a Global Positioning System (GPS) receiver, to name just a few.

Computer readable media suitable for storing computer program instructions and data include all forms of non volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and CD ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

To provide for interaction with a user, embodiments of the subject matter described in this specification can be implemented on a computer having a display device, e.g., a CRT (cathode ray tube) or LCD (liquid crystal display) monitor, for displaying information to the user and a keyboard and a pointing device, e.g., a mouse or a trackball, by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input.

Embodiments of the subject matter described in this specification can be implemented in a computing system that includes a back end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the subject matter described in this specification, or any combination of one or more such back end, middleware, or front end

components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network (“LAN”) and a wide area network (“WAN”), e.g., the Internet.

5 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.

10 While this specification contains many specific implementation details, these should not be construed as limitations on the scope of any invention or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular inventions. Certain features that are described in this specification in the context of 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
15 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.

20 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
25 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.

30 Particular embodiments of the subject matter described in this specification have been described. Other embodiments are within the scope of the following claims. For example, the actions recited in the claims can be performed in a different order and still achieve desirable results. As one example, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable

results. In certain implementations, multitasking and parallel processing may be advantageous.

CLAIMS

1. A computer-implemented method, comprising:

receiving composition inputs in an input method editor environment, the composition inputs based on user inputs;

5 identifying candidate selections in the input method editor environment based on the composition inputs, each candidate selection comprising one or more ideograms;

receiving in the input method editor environment a search command, the search command based on a user input;

selecting one of the candidate selections in response to the search command; and

10 providing the selected candidate selection and the search command to a search engine interface external to the input method editor environment.

2. The method of claim 1, wherein providing the selected candidate selection and the search command to a search engine interface comprises:

15 in response to the search command, determining if the search engine interface is active;

if the search engine interface is determined to be active:

providing the selected candidate selection to the search engine interface as a search input; and

20 invoking a search in the search engine interface; and

if the search engine interface is determined to be inactive:

instantiating the search engine interface;

providing the selected candidate selection to the search engine interface as a search input; and

25 invoking a search in the search engine interface.

3. The method of claim 1, wherein:

identifying candidate selections in the input method editor environment based on the composition inputs comprises identifying Hanzi characters based on one or more Pinyin inputs; and

30 providing the selected candidate selection and the search command to a search engine interface comprises providing selected Hanzi characters as a search input to the search engine interface.

4. The method of claim 1, wherein selecting one of the candidate selections in response to the search command comprises manually selecting one of the candidate selections.

5. The method of claim 1, comprising:
ranking the candidate selections in the input method editor;
presenting the candidate selections in the input method editor environment according to the rank; and
wherein selecting one of the candidate selections in response to the search command
comprises:
automatically selecting a first ranked candidate selection.

6. The method of claim 1, wherein the composition inputs comprise keyboard inputs.

7. A computer-implemented method, comprising:
storing in a input method editor data store integration commands, each integration command identifying an integration of the input method editor with a corresponding application environment that is external to the input method editor user interface;
identifying candidate selections in an input method editor user interface, the candidate selections based on one or more composition inputs;
receiving in the input method editor user interface an integration command, the integration command based on a user input and corresponding to one of the store integration commands;
selecting one of the candidate selections in response to the integration command; and
providing the selected candidate selection to the corresponding application environment external to the input method editor user interface.

8. The method of claim 7, wherein providing the selected candidate selection to the corresponding application environment external to the input method editor user interface comprises:

in response to the integration command, determining if the application is active;
if the application is determined to be active:
providing the selected candidate selection to the application; and

invoking an application function that utilizes the selected candidate selection as input; and

if the application is determined to be inactive:

instantiating the application;

5 providing the selected candidate selection to the application; and

invoking an application function that utilizes the selected candidate selection as input.

9. The method of claim 8, wherein:

10 identifying candidate selections in an input method editor user interface, the candidate selections based on one or more composition inputs comprises identifying Hanzi characters based on one or more Pinyin inputs; and

providing the selected candidate selection and the integration command to the corresponding application interface comprises providing selected Hanzi characters as a search
15 input to a search engine interface.

10. The method of claim 8, wherein:

the application comprises a search engine user interface; and
the application function comprises a search operation.

20

11. The method of claim 8, wherein:

the application comprises an e-mail application; and
the application function comprises an e-mail composition function.

25 12. The method of claim 7, comprising:

identifying a context defined by candidate selections; and
identifying integration commands related to the identified context; and
enabling the identified integration commands.

30 13. The method of claim 7, comprising:

simultaneously presenting a plurality of integration command activation elements.

14. The method of claim 7, wherein each candidate selection comprises one or more ideograms

15. A system comprising:

means for storing in a input method editor data store integration commands, each integration command identifying an integration of the input method editor with a
5 corresponding application environment that is external to the input method editor user interface;

means for identifying candidate selections in an input method editor user interface, the candidate selections based on one or more composition inputs;

means for receiving in the input method editor user interface an integration command,
10 the integration command based on a user input;

means for selecting one of the candidate selections in response to the integration command; and

means for providing the selected candidate selection and the integration command to the corresponding application environment external to the input method editor user interface.
15

16. A system, comprising:

a data store storing one or more input method editor integration commands, each integration command identifying an integration of an input method editor with a corresponding application instance that is external to an input method editor user interface;

20 a plurality of application interfaces, each application interface associated with one of the corresponding application instances and configured to facilitate data communication from the input method editor to the application instance; and

an input method editor configured to:

identify candidate selections based on one or more composition inputs; and

25 in response to receiving an integration command associated with an application interface, access the application interface, select one of the candidate selections and provide the selected candidate selection to the corresponding application instance associated with the application interface.

30 17. The system of claim 16, wherein each application interface is configured to instantiate its corresponding application instance in response to the input method editor receiving an integration command associated with the application interface.

18. The system of claim 17, wherein the candidate selections are Hanzi characters, and the composition inputs are Pinyin inputs.

19. The system of claim 17, wherein the selected candidate selection is a user selected candidate selection.

20. The system of claim 16, wherein the input method editor is further configured to:
identify contexts defined by candidate selections;
identify integration commands related to the identified contexts; and
enable the identified integration commands.

21. The system of claim 20, wherein the identified context is a city context, and the integration comment related to the identified context is a weather data comment.

22. The system of claim 20, wherein the identified context is a stock context, and the integration comment related to the identified context is a financial data comment.

1/3

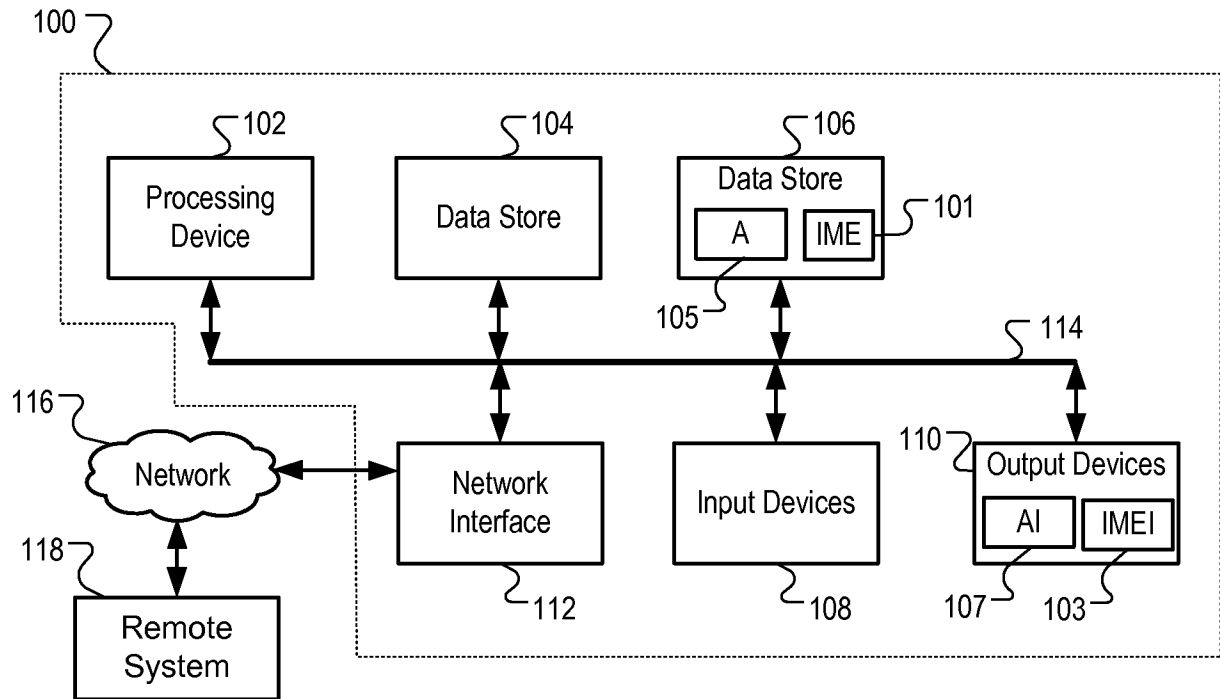


FIG. 1

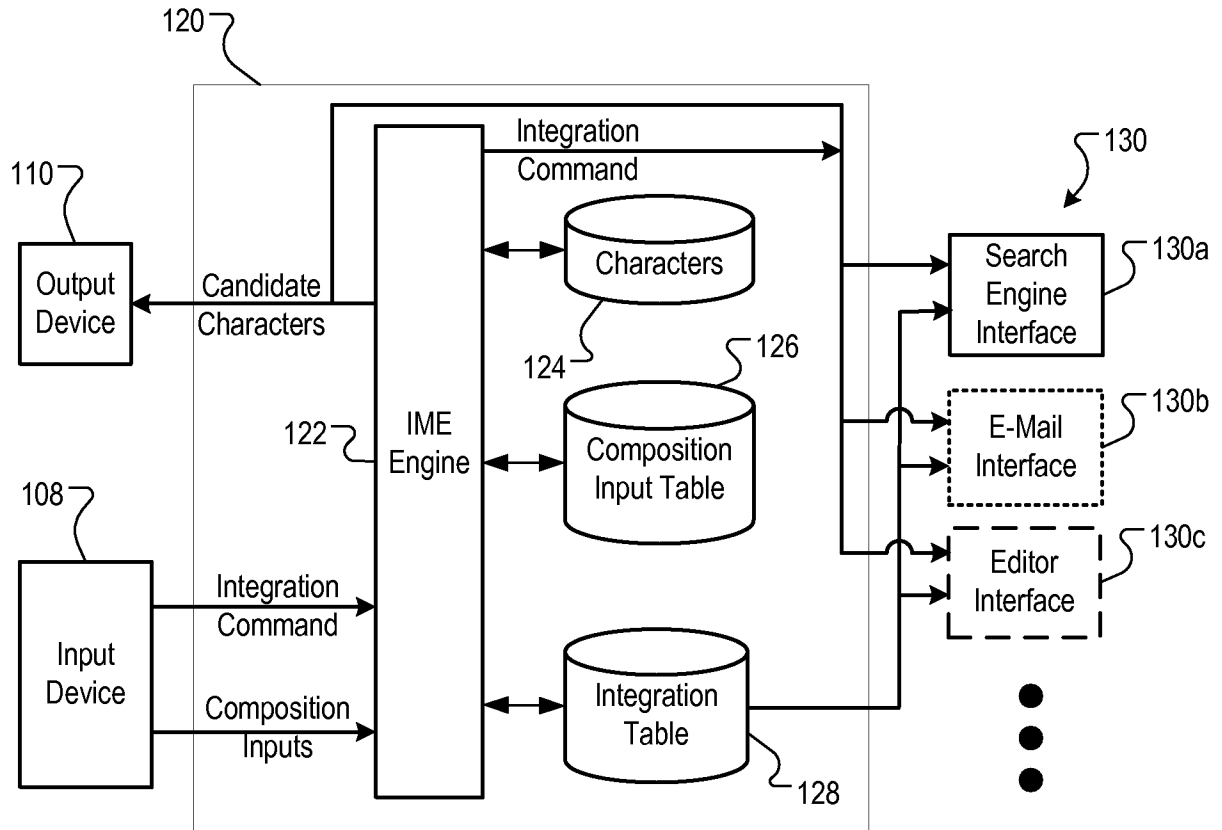
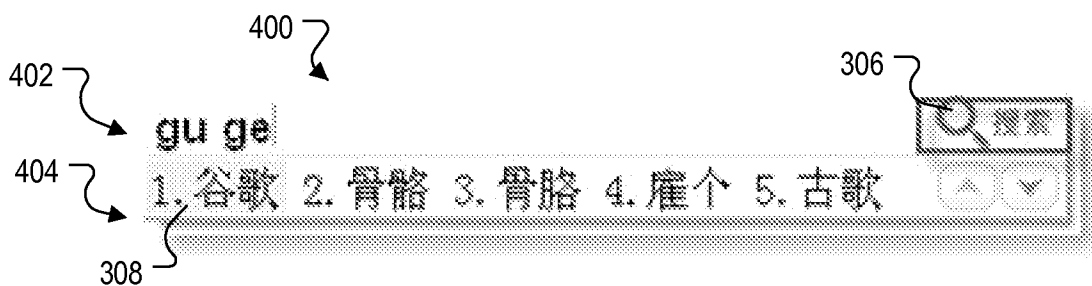


FIG. 2

2/3



3/3

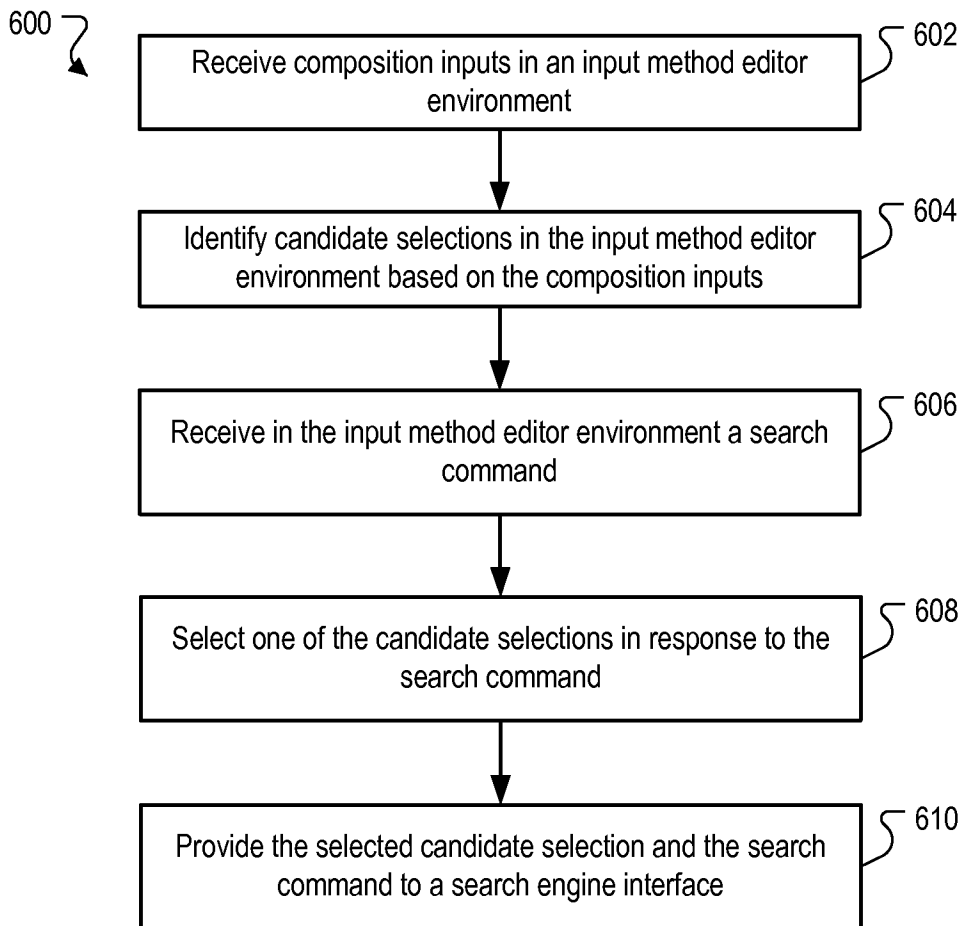


FIG. 6

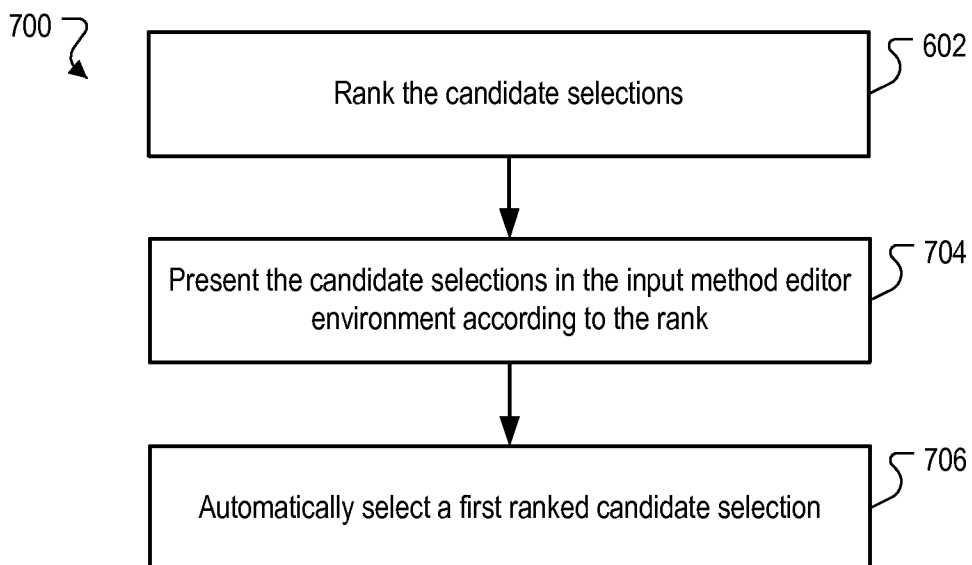


FIG. 7

A. CLASSIFICATION OF SUBJECT MATTER**G06F 17/28(2006.01)i, G06F 17/00(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)

IPC8 : G06F

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

Korean Utility models and applications for Utility models since 1975

Japanese Utility models and applications for Utility models since 1975

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

eKIPASS(KIPO internal) "input", "editor", "IME", "search", "candidate"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7,174,288 B2 (YUN-CHENG JU, et al.) 6 February 2007 see abstract, column 2 lines 10-45, and figures 1-9.	1, 7, 15, 16
A		2-6, 8-14, 17-22
A	US 7,099,876 B1 (DAVID JAMES HETHERINGTON, et al.) 29 August 2006 see abstract, column 2 line 48 - column 3 line 17, and figures 2-7.	1-22
A	US 5,432,948 A (MARK E. DAVIS, et al.) 11 July 1995 see abstract, column 2 line 66 - column 6 line 53, and figures 1-6.	1-22



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

27 AUGUST 2008 (27.08.2008)

Date of mailing of the international search report

27 AUGUST 2008 (27.08.2008)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 139 Seonsa-ro, Seo-
gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

LEE, Seok Hyung

Telephone No. 82-42-481-8507



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2008/059760

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7174288 B2	06.02.2007	CN 1457002 A JP 2004-046807 A2 US 2003-212563 A1	19.11.2003 12.02.2004 13.11.2003
US 7099876 B1	29.08.2006	None	
US 5432948 A	11.07.1995	AU 1994-60195 A1 CA 2145668 AA DE 69400869 C0 DE 69400869 T2 EP 0686286 B1 EP 0686286 A1 JP 08-509829 A WO 94-25922 A1	21.11.1994 10.11.1994 12.12.1996 15.05.1997 06.11.1996 13.12.1995 15.10.1996 10.11.1994