USER INTERFACE METHOD AND APPARATUS

FIELD OF THE INVENTION

[0001] The present invention relates to a user interface method and apparatus and in one aspect to a user interface method and machine to enable at least one user to enter an input to execute a command to transition a machine from one state to another.

BACKGROUND INFORMATION

[0002] Electronic processing devices including computers, media devices, and mobile devices are used prevalently in almost all facets of people's lives. The ease with which users can interact with these devices is the subject of considerable research and commercial effort. Making the user interface of electronic machines easier for users is of major importance and can make a very powerful and useful apparatus difficult to use of even useless in the eyes of a user.

[0003] For example, in the field of web browsing it is sometime necessary for a user to type in the text characters comprising an HTTP request in order to access a web page. The syntax of HTTP is not intuitive to many users. The input of the text can be subject to error and hence users can experience frustration with the inability to access the correct web page accurately. This problem has been partly addressed by the use of predictive text suggestions, such as by enabling Bing (trade mark of Microsoft Corporation) to search for potential HTTP addresses corresponding to the characters as entered by the user. However, this function requires the characters to be transmitted to a server to carry out a search for likely results. The search in not performed with any context on what the user might want to find apart from using the entered characters.

SUMMARY OF THE INVENTION

[0004] One aspect provides a method, in a programmable machine, of using a user interface to cause commands to be executed by the machine, the method comprising storing shortcut data for at least one user, the shortcut data comprising a plurality of sets of identifiers for shortcuts for a plurality of states of the machine for the at least one user, each stored identifier for a shortcut having a corresponding stored machine command, the state of the machine comprising a stage of processing of the machine; determining a current state of the machine for a said user; receiving an input from the user; determining whether the input

matches a stored identifier for a shortcut in the shortcut data; executing a stored machine command corresponding to the stored identifier dependent upon the matching to transition the machine to a next state of the machine; updating stored history data for the at least one user; and updating the shortcut data using the updated history data.

[0005] Another aspect of the invention provides a programmable machine providing a user interface for causing the execution of commands by the machine, the machine comprising a program memory storing program code; and a processor for implementing the program code stored in the program memory; wherein the program code comprises: code for controlling the processor to carry out the method described above.

[0006] Another aspect of the invention provides a carrier medium carrying processor executable code, which when executed by a processor of a machine, causes the machine to carry out the method described above.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Figure 1 is a schematic diagram illustrating a user device as a machine according to one embodiment;

[0008] Figure 2 is a flow diagram illustrating the process according to one embodiment;

[0009] Figure 3 is a schematic diagram illustrating a user device as a machine synchronizing history data and shortcut data with a remote server according to one embodiment;

[0010] Figure 4 is a schematic diagram illustrating a server as a multiuser machine according to one embodiment;

[0011] Figures 5a and 5b are illustrations of user interface outputs according to one embodiment;

[0012] Figure 6 is a schematic diagram illustrating a server as a multiuser machine synchronizing history data and shortcut data with a remote server according to one embodiment; and

[0013] Figure 7 is a schematic diagram of a basic computing device for use in one embodiment.

DETAILED DESCRIPTION

[0014] In the following detailed description, reference is made to the accompanying drawings that form a part hereof, and in which is shown by way of illustration specific embodiments in which the inventive subject matter may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice them, and it is to be understood that other embodiments may be utilized and that structural, logical, and electrical changes may be made without departing from the scope of the inventive subject matter. Such embodiments of the inventive subject matter may be referred to, individually and/or collectively, herein by the term "invention" merely for convenience and without intending to voluntarily limit the scope of this application to any single invention or inventive concept if more than one is in fact disclosed.

[0015] The following description is, therefore, not to be taken in a limited sense, and the scope of the inventive subject matter is defined by the appended claims.

[0016] In the following embodiments, like components are labelled with like reference numerals.

[0017] In the following embodiments, data is described as being stored in at least one database. The term database is intended to encompass any data structure (and/or combinations of multiple data structures) for storing and/or organizing data, including, but not limited to, relational databases (e.g., Oracle databases, mySQL databases, etc.), non-relational databases (e.g., NoSQL databases, etc.), in-memory databases, spreadsheets, as comma separated values (CSV) files, eXtendible markup language (XML) files, TeXT (TXT) files, flat files, spreadsheet files, and/or any other widely used or proprietary format for data storage. Databases are typically stored in one or more data stores. Accordingly, each database referred to herein (e.g., in the description herein and/or the figures of the present application) is to be understood as being stored in one or more data stores. A "file system" may control how data is stored and/or retrieved (for example, a disk file system like FAT, NTFS, optical discs, etc., a flash file system, a tape file system, a database file system, a transactional file system, a network file system, etc.). For simplicity, the disclosure is described herein with respect to databases. However, the systems and techniques disclosed herein may be implemented with file systems or a combination of databases and file systems.

[0018] In the following embodiments, the term data store is intended to encompass any computer readable storage medium and/or device (or collection of data storage mediums and/or devices). Examples of data stores include, but are not limited to, optical disks (e.g., CD-ROM,

DVD-ROM, etc.), magnetic disks (e.g., hard disks, floppy disks, etc.), memory circuits (e.g., solid state drives, random-access memory (RAM), etc.), and/or the like. Another example of a data store is a hosted storage environment that includes a collection of physical data storage devices that may be remotely accessible and may be rapidly provisioned as needed (commonly referred to as “cloud” storage).

[0019] The functions or algorithms described herein are implemented in hardware, software or a combination of software and hardware in one embodiment. The software comprises computer executable instructions stored on computer readable media such as memory or other type of storage devices. Further, described functions may correspond to modules, which may be software, hardware, firmware, or any combination thereof. Multiple functions are performed in one or more modules as desired, and the embodiments described are merely examples. The software is executed on a digital signal processor, ASIC, microprocessor, or other type of processor operating on a system, such as a personal computer, server, a router, or other device capable of processing data including network interconnection devices.

[0020] Some embodiments implement the functions in two or more specific interconnected hardware modules or devices with related control and data signals communicated between and through the modules, or as portions of an application-specific integrated circuit. Thus, the exemplary process flow is applicable to software, firmware, and hardware implementations.

[0021] A generalized embodiment provides a machine and a method for providing a user interface using stored shortcut data for at least one user or in other words a set of one or more users. The shortcut data comprises a plurality of sets of identifiers for shortcuts for a corresponding plurality of states of the machine for the set of one or more users or a plurality of sets of states for the set of one or more users. Each stored identifier for a shortcut has a corresponding stored machine command. A current state of the machine used by a user is determined and when an input is received from the user, it is determined whether the input matches a stored identifier for a shortcut in the shortcut data. If it does, a stored device command corresponding to the stored identifier is executed, the stored history data for the set of one or more users is updated and the shortcut data is updated using the updated history data.

[0022] The machine can hence store history data to learn about user inputs for specific machine states for use in the determination of updated shortcut data to store appropriate

shortcuts for commands that a set of one or more users might want to enter in a current machine state to transition to a next machine state. Hence, the shortcuts are user and context specific.

[0023] The user interface method can be used to learn inputs for a set of users using the same machine states e.g. using the same applications, programs or processes. The set of users can be any number including a single user. The user interface method can be used by a group of users, such as an office or business.

[0024] The invention is applicable to any electronic machine having a need to receive user inputs to enter commands to the machine. In a programmable machine such as a computer, the user interface can monitor any form of user input such as from a keyboard, pointer device, touch screen input, audio or speech input, or gesture input. The input can also comprise an image obtained by scanning or photographing an object or by downloading an image over a network from an image source. Where the input is an image, the matching of the input to identifiers can comprise matching an input image to a stored image associated with a command or command sequence. The match can be exact or based on a level of similarity.

[0025] The user interface can be software agnostic in that it can monitor the input to any application operated by an operating system or even an input to the operating system itself. The machine can thus provide a multimodal interface to the user or users.

[0026] Where one machine learns about user behaviour from user inputs or from other user behaviour, that machine can determine shortcut data to store appropriate commands that a user might want to enter in a current machine state to transition to a next machine state. This stored shortcut data can be transferred to another machine having the same machine states used by the same user so that the shortcut data is available to make that machine easier to use.

[0027] In one embodiment, if the input from the user does not match a stored identifier, it is determined whether the input is a command, and if so the command is executed, the stored history data for the user is updated, and the shortcut data using the updated history data is updated. In this way, in an initial condition, where there is no history data learnt from the set of user's previous inputs, the user's input cannot match any shortcut identifier and so the machine can update the history data to start the learning process. This learning process of updating the history data can continue, whenever the users enter inputs that are not a shortcut identifier. The history data is updated and the updated history data can be used to make better-suggested shortcuts for the set of users.

[0028] The shortcut identifiers may or may not be output to the set of users for the current state of the machine. The shortcut identifiers can be output for example visually on a

display, audibly by an audio output or as a vibrational output. The output of the shortcut identifiers assists the set of users in making the selection of a shortcut to enter a command. This avoids the need for a user to remember the shortcuts. However, the user may find that the output of the shortcut identifiers becomes a nuisance and a distraction from the output from the machine in the current state. For example, the display of selectable shortcut identifiers as buttons or icons may be considered by a user to be visually distracting while they are viewing the display for the current state of the machine.

[0029] The shortcut identifiers may be output in an order dependent on the history data. This can allow a user to more easily select a shortcut that they use more frequently. For example, in an embodiment, whether the shortcut identifiers are displayed, the history data enables the determination of commands and hence shortcuts that are input by the set of users more frequently or more recently than others, so that they can be displayed either more prominently or at the highest position in a list or ordered sequence. In an audio output embodiment, shortcut identifiers for the more frequently used or more recently used commands can be audibly output first to assist the set of users more quickly make an audio selection/input of a shortcut identifier.

[0030] The stored machine command can comprise a single command or a plurality of sequential commands to be executed by the machine sequentially. The storage of a sequence of commands can be determined as a shortcut by learning a repeated pattern of user input by the user from the history data. This pattern can then be stored as a command sequence in a single shortcut.

[0031] In one embodiment, the stored shortcut data and the shored history data for the set of users can be synchronized with a remote database of shortcut data and history data for a plurality of users including the set of users. Thus in this embodiment, a remote database can store shortcut data and history data for a plurality of users so that it can be remotely processed. For example, in one embodiment, the history data received from the remote database in the synchronization includes user preference data indicating preferences of the set of users, and the shortcut data is updated using the user preference data. User preference data can be gathered from data collected from or about users from many sources, such as merchant loyalty programs and merchant transaction data. The data on user transaction behavior can be processed to determine user preferences.

[0032] The machine can be a single user machine, a standalone machine, or a multiuser machine such as a server and the shortcut data can be stored for a single set of users or a

plurality of sets of users of the machine. For multiple sets of users, multiple sets of history data and shortcut data has to be stored: one set per set of users. In one embodiment, the stored shortcut data and the shored history data for the users can be synchronized with a remote database of shortcut data and history data for a plurality of sets of users including the users of the machine. Thus in this embodiment, a multiuser machine can store data for multiple sets of users, such as customers of a vendor or merchant or multiple businesses, and a remote database can store history data and shortcut data for the plurality of sets of users (e.g. for a plurality of vendors or businesses) so that it can be remotely processed. For example, in one embodiment, the history data received from the remote database in the synchronization includes user preference data indicating preferences of the users, and the shortcut data is updated using the user preference data. User preference data can be gathered from data collected from or about a set of users from many sources, such as merchant loyalty programs and merchant transaction data. The data on user transaction behaviour can be processed to determine user preferences. Embodiments also encompass the management of financial instruments, such as money or currency. Payment or spending preferences or habits of a user can be learnt to provide shortcuts for machine states to next machine states, such as allowing a user to make payments.

[0033] In one embodiment, the input from the user can include one or more text characters in a displayed text entry field, the stored identifier for the shortcut can include a plurality of text characters, and the corresponding command can include the plurality of text characters. In order to determine whether the input matches a stored identifier for a shortcut in the shortcut data, the input one or more text characters are matched with one or more of the plurality of text characters of the stored identifier.

[0034] This embodiment of the invention enables the invention to be applied to a text field in a user interface. As a user enters characters in the text field, the characters are compared with shortcut identifiers in the form of text for commands in order to output suggested shortcuts to the commands based on the current state of the machine and the user's personal input history. For example, when a user interface displays a web page, the user's history will indicate that the user made certain inputs previously when on that web page, and hence the shortcut data reflects those inputs with shortcut identifiers for inputs that the user is more likely to make.

[0035] In one embodiment, the user interface can be used in an electronic transaction the method disclosed in co-pending application number PCT/GB2016/054089, filed the same day as this application, entitled "AN ELECTRONIC TRANSACTION METHOD AND

APPARATUS", by the same applicant and inventors (attorney reference 11043PUSA), the content of which is hereby incorporated in its entirety.

[0036] Specific embodiments will now be described with reference to the drawings.

[0037] A first embodiment will be described with reference to figures 1 and 2. Figure 1 illustrates a device 100 for use by a set of one or more users not connected to any network (a standalone machine). The set of users can be any number including a single user. The user interface method can be used by a set of users, such as an office or business, using the same machine states e.g. using the same applications, programs or processes, such that one of the set of users is likely to want to enter a command in a machine state in the same manner as any other member of the set of users.

[0038] The device 100 can comprise any type of computing or processing machine, such as a personal computer, a laptop, a tablet computer, a personal organizer, a mobile device, smart phone, a mobile telephone, a video player, a television, an audio player, a multimedia device, personal digital assistant, etc.

[0039] A processor 20 is provided to load program code 21 from a program code store 70 and execute the program code 21 using working volatile memory 30. The program code 21 executes the functionality of one embodiment of the invention. Other program code can be loaded up in the form of an operating system, and applications to receive the commands that a user inputs directly or inputs by inputting a shortcut for a command in accordance with one embodiment.

[0040] A user interface 5 comprises an input device 10 and an output device 40. The input device can comprise any hardware, software or combination of hardware and software to receive a user input, such as a keyboard, touch screen, pointer device (such as a mouse, trackball, trackpad, or pen device), audio recognition hardware and/or software to recognize a sounds or speech from a user, gesture recognition input hardware and/or software, etc. The output device can comprise any hardware, software or combination of hardware and software to output information to a user including shortcut identifiers, such as a display screen, or an audio speaker. Hence, the user interface 5 provides for outputs to a user from the processor 20 and inputs from the user to the processor 20.

[0041] The device 100 is also provided with a data store 60 to store shortcut data in a shortcut database 61 and history data in a history database 62. Although the shortcut data and

the history data are described as being stored in databases in this embodiment, the data can be stored in any convenient form, or data structure.

[0042] The history data for a set of one or more users comprises a set of data for the set of one or more users for each of a set of states of a machine used by the set of one or more user. For each state of the machine a set of one or more user input commands are stored to record the commands directly input or input as a result of a user selection (e.g. clicking a displayed selectable option such as a hyperlink in an interface). In this way, the history data can record for each set of one or more users, and for each machine state used by the set of one or more users, a set of input commands.

[0043] In one embodiment, the history data can also include user preference data indicating preferences of a user. The user preference data can be obtained from analyzing the user inputs and processing other user data entered by a user that can indicate a user's preferences. The preference data can be used along with the recorded user input history data to determine updates to the shortcut data.

[0044] The shortcut data stores, for each machine state, a shortcut identifier and one or more commands to be executed by the machine. If there is more than one command, the commands are to be executed sequentially. The shortcut identifier can comprise a tag that a user can use to simply select the shortcut e.g. 1, 2, 3, 4…. or A, B, C, ….. This tag can be automatically assigned to the shortcut or a user can select a tag. The tag enables the user to make an input simply by inputting the tag e.g. 2, or D, rather than typing some form of text string. The shortcut data can also comprise information for display to the user when the shortcut identifier is displayed for selections as for example, and icon, image, button etc. The information can assist the user in identifying the command available for selection as a shortcut.

[0045] Alternatively, the shortcut identifier can comprise a text string corresponding to the text command string entered as the input by the user e.g. an http input command to load a web page into a browser, the name of a file to loaded in an application or to be searched for in a file system, or the name of an application to be executed in an operating system. In this example, the matching of a user's input can be achieved by matching the text characters as they are typed in by a user to suggest potential commands that the user might be trying to type in based on previously type commands. The user input text can be matched on a partial match basis with the stored shortcut identifier text so that initially a number of possible matches are identified that the user can select from and these are reduced as the user types more characters so that some initial shortcut identifier text matches no longer match and are removed as options

for the user. This is a form of predictive text, but it is user and context specific in that it is specific to the user and the current state of the machine e.g. the web page displayed by a browser used by the user, a document opened by a user, a search window opened for a document in an operating system, search engine, etc.

[0046] In embodiments of the present invention, a machine state can comprise any operating state currently instantiated by the user of the machine. In a programmed device, the state can comprise a state in the operation of a computer program, such as a web page in a web browser application, a document in a word processor application, a worksheet in a spreadsheet application, an email in an email application, operating system navigation and search features, or any search and retrieval application.

[0047] The operation of one embodiment will now be described with reference to the flow diagram of figure 2.

[0048] In step S10 the current state of the machine is determined, e.g. the current web page open in a browser, the current document open in the word processor, the current email open in an email application, etc. Based on the current machine state, the process determined whether there are any stored shortcuts available in step S11. In one embodiment, the available shortcuts can only include shortcuts that were recently created or used. Shortcuts that have not been used for a long period of time are likely not to be relevant to a user and can hence be deleted to avoid presenting the user with out of date old shortcuts. If there are available shortcuts determined to be available in step S11, in step S16, the process determines whether the shortcuts are to be output to the user for selection. If the output of the shortcuts is not enabled or disabled by the user (for example because they have become familiar with the available shortcuts and do not need or want them to be output e.g. displayed), the process proceeds to step S17 to await an input from the user. If the shortcuts are determined to be output in step S16, the process proceeds to step S15 to output the shortcut options to the user, e.g. display shortcut icons, buttons, or images with shortcut identifiers and optionally some shortcut information to assist the user make the shortcut selection. The process then proceeds to step S17 to await an input from the user.

[0049] If in step S11 it is determined that there are no available up to date shortcuts, a shortcut determination process S30 is carried out comprising looking up history data for the user in step S12, determining shortcuts for the user for the current machine state using the history data in step S13, and storing the shortcut data in step S14. The stored shortcuts are then output to the user in step S15, e.g. display shortcut icons, buttons, or images with shortcut

identifiers and optionally some shortcut information to assist the user make the shortcut selection. The process then proceeds to step S17 to await an input from the user.

[0050] When an input is received from the user in step S17, the input from the user is used in step S18 to look up a shortcut for the current state to determine one or more commands to be executed by the machine. The input from the user can be an input of the shortcut identifier e.g. by a text input or a voice input, of a selection of a shortcut option, whereby the selection is converted to a user input for comparison with the shortcut identifiers for the current machine state.

[0051] In step S19 it is determined whether the input from the user matches a shortcut. If the set of one or more users have not made the input at the current machine state before so that it is not in the history data for use in the formation of a shortcut, in step S23 the process determines whether the input is a command input or merely some other form of input, such as an input of text into a document, of other input field that does not cause the machine to transition to another state. For example, in a web form, data can be entered without it being sent to a server and hence the entered text data is not a command. When it is sent to the server it is a command since the web page will be usually be updated. If it is determined in step S23 that the user input is not a command, the process returns to step S17 to await a user input.

[0052] If a command is determined to be input in step S23, the process moves to step S24 to execute the command to transition the machine to the next state. Then in step S21, the history data for the previous machine state is updated with the command to transition to the next state. In step S22, the updated history data is then used to generate updated shortcuts and the process returns to step S10 to determine the current machine state.

[0053] If in step S19 it is determined that the input does match a shortcut, in step S20, the command for the shortcut is executed to take the machine to the next start. Then in step S21, the history data for the previous machine state is updated with the selection of the shortcut to transition to the next state. In step S22, the updated history data is then used to generate updated shortcuts and the process returns to step S10 to determine the current machine state.

[0054] In the updating of the history data, when a new command is entered by the user a new shortcut can be created. When a user selects a previously created shortcut, the user’s selection of the shortcut can be recorded in the data so that the relevance of shortcut can be increased and the shortcut is indicated as being recent and not old. Selections of previously created shortcuts can be used to rank or order the shortcuts for output to the set of one or more

users so that the most relevant shortcuts are output and they are ordered according the relevance i.e. how often selected and how recently.

[0055] Although the shortcut determination process S30 is illustrated in figure 2 as being performed every time there are no available up to date shortcuts, in one embodiment, the steps of the shortcut determination process S30 can be performed for each machine state for the set of one or more users in a separate or 'offline' process by the processing of the history data for each machine state. Where the history data include user preference data, even if any user of the set of users has not made an input to transition from a machine state, in one embodiment, one or more shortcuts can be determined based solely on the user preference data, obtained for example from user behavior using the machine or from user input parameters.

[0056] Figure 3 illustrates another embodiment of the present invention in which a remote database of shortcut and history data for a plurality of sets of one or more users is provided.

[0057] In this embodiment, the user's device 1 comprises the components illustrated and described with reference to figure 1. The user's device 1 operates in the same manner as described with reference to the flow diagram of figure 2. In addition, the user's device 1 includes a network interface 50 for connecting the user's device1 over a network such as the internet 80 to a remote server system 90. The server system 90 is connected to or provided with a data store 95 in which are stored a global shortcut database 96 and a global history database 97.

[0058] The global shortcut database 96 stores shortcuts for the set of one or more users of the user's device 1 and a plurality of other sets of users of other user's devices. Similarly, the global history database 97 stores the history for the set of users of the user's device 1 and a plurality of other sets of users of other user's devices. The data stored in the data store 95 can be synchronized with the data stored in the data store 60 in the user's device. In this way, in the embodiment where user preference data is included in the history data, the shortcuts and history stored in the data store 95 can reflect user preferences and can be sent to the data store 60 on the user's device 1.

[0059] Figure 4 illustrates another embodiment comprising a server as a multiuser machine according to one embodiment.

[0060] In this embodiment, a server system 90 includes a web server 92 providing access over a network such as the internet 80 to user's devices 1 operating respective web

browsers 2. The server system 90 also executes an application server 91 and accesses a data store 95 storing a global shortcut database 96 and a global history database 97.

[0061] In this embodiment, the server system 90 operates according to the flow diagram of figure 2 for each user of a plurality of users accessing the web server 92 by accessing user specific sets of data for machine states comprising states of the server system 90 in the global shortcut database 96.

[0062] Figures 5a and 5b illustration user interface outputs for the server system of figure 4 according to one embodiment. In this embodiment, the web server 92 serves web pages to be rendered in the browser 2 of each user's device 1. The user interface thus comprises a displayed web page 400 comprising a universal resource locator (URL) text entry field 402 in a header region 401. The web page displayed in figure 5a comprises a search engine web page, called "Super web searcher" and a text search input field 404 is provided to allow a user to enter a text string as a search string to send as part of a command to undertake a web search. In this embodiment, the user has input the text string "http://spee" in either the URL text entry field 402 or the text search input field 404 (both are shown in figure 5a, but in practice the user will only input text to one of the fields 402 and 404). As a result of the user input text, the user input is matched to shortcut identifiers for shortcuts and shortcut icons, windows, drop down lists, buttons or images 403 are displayed. The shortcuts 403 displayed comprise:

Shortcut identifier	Command
1	The URL for Speedy Shop web page
2	The URL for Speedway track web page
3	The URL for Speed Savers web page

[0063] In this embodiment, on this web page (machine state), the user has only made user input comprising text inputs and hence the shortcuts can be selected by inputting initial characters of the text string previously entered and stored in the history.

[0064] The shortcuts 403 can display an image or information on the command that will be entered if the shortcut 403 is selected to assist the user identify the command entry options available using the shortcuts. The identifier in this embodiment, comprises a short label or tag so that the selection of the shortcut can be made by any one of: using a pointer device, gesture, or touch screen to select a displayed location of a shortcut 403, using a keyboard to input 1, 2 or 3, or using speech recognition to speak 1, 2 or 3 as the input.

[0065] Figure 5b illustrates the web page 400a displayed as a result of the use selection of shortcut tag 1 in figure 5a.

[0066] In this embodiment, the web page 400a is displayed for “Speedy Shop”, a vendor or merchant of electrical goods. The web page displays conventional hyperlinks to open web pages for various product types, such as camera, laptops, televisions, and loud speakers. In the header 401 is provided the URL text field 402 as well as shortcuts 403a. In this embodiment, the shortcut 403a are displayed in the header region 401 so as to leave the rest of the display area free and avoid the shortcuts overlapping web page information.

[0067] The shortcuts 403a displayed comprise:

Shortcut identifier	Command
1	The URL link to the Laptops web page
2	The URL and sequential commands to take the user directly to the vendor’s checkout with a specific camera that’s in the user’s basket
3	The URL link to the Speakers web page

[0068] It can thus be seen that in this embodiment, that the shortcuts are identified by a tag separately to any input text by a user. In this machine state a user has previously made selections of the commands stored in the shortcuts and hence the shortcuts can simply be selected by selecting a tag

[0069] Figure 6 illustrates a server as a multiuser machine synchronizing history data and shortcut data with a remote server according to one embodiment. In this embodiment, each vendor system 90A and 90B can comprise a server system 90 as described with reference to figure 4. Each vendor server 90A and 90B operates substantially as described in the flow diagram of figure 2 for each user accessing a web server 92A or 92B of a respective vendor server 90A or 90B using a web browser 2 of a respective user’s device 1 over a network such as the internet 80. Each vendor server system 90A and 90B includes an application server 91A and 91B, a data store 95A and 95B storing a respective vendor shortcut database 96A and 96B and a respective vendor history database 97A and 97B. In this embodiment, the databases 96A, 96B, 97A and 97B store data for a plurality of users of the vendor’s web site.

[0070] A third party provider server system 190 is connected over the internet 80 to the vendor server systems 90A and 90B. The third party provider server system 190 includes an application serve 191 accessing a global shortcut database 196 and a global history database 197. The global databases 196 and 197 store data for a plurality of users for a plurality of vendor server systems 90A and 90B. Thus in a similar manner to the data store 95 of the embodiment of figure 3 holding data for a plurality of user's devices 1 of figure 1, in this embodiment the databases 196 and 197 hold data for sets of users of a plurality of vendor server systems 90A and 90B.

[0071] The global shortcut database 196 stores shortcuts for the users of the user's devices 1 that use the vendors server system 90A and 90B and a plurality of other users of other user's devices that use other vendor server systems. Similarly, the global history database 197 stores the history for the users of the user's devices 1 that use the vendors server system 90A and 90B and a plurality of other users of other user's devices that use other vendor server systems. The data stored in the databases 196 and 197can be synchronized with the data stored in the data store 95A and 95B in the vendors server system 90A and 90B. In this way, in the embodiment where user preference data is included in the history data, the shortcuts and history stored in the data bases 196 and 197 can reflect the user preferences determined from multiple vendors and other sources and can be sent to the data store 95A and 95B on the vendor server system 90A and 90B. This allows vendors to share user preference data that can be determined from multiple vendor data that can include the history data recorded for machine state transitions by users using the vendors server systems.

[0072] Although embodiments have been described with reference to the machine state comprising a web page served by a web server, the machine state can comprise any process step implemented in a processing machine that a user can make an input into to cause a transition to a next machine state. For example, a machine state can be represented by a stage of processing in any software application, such as a document, a spreadsheet, email, operating system navigation or input, and searching and retrieval software.

[0073] Embodiments provide for context based options for input of user selection options as a graphic selection or an audio selection/input to provide for predictive user input.

[0074] In one embodiment, a current machine state can be a set of machine states such as web pages or other interface displays having common parameters. For example, a web page can be scrolled or zoomed, and frames can hold data entered by the user such as name and

contact details. However, such data is handled by the web browser and is not transmitted to the web server and hence the machine state of the client device may operate in multiple states having common parameters that are only a single web page serving step for the web server.

BASIC COMPUTING DEVICE

[0075] Figure 7 is a block diagram that illustrates a basic computing device 600 in which the example embodiment(s) of the present invention may be embodied. Computing device 600 and its components, including their connections, relationships, and functions, is meant to be exemplary only, and not meant to limit implementations of the example embodiment(s). Other computing devices suitable for implementing the example embodiment(s) may have different components, including components with different connections, relationships, and functions.

[0076] Computing device 600 may include a bus 602 or other communication mechanism for addressing main memory 606 and for transferring data between and among the various components of device 600.

[0077] Computing device 600 may also include one or more hardware processors 604 coupled with bus 602 for processing information. A hardware processor 604 may be a general purpose microprocessor, a system on a chip (SoC), or other processor.

[0078] Main memory 606, such as a random access memory (RAM) or other dynamic storage device, also may be coupled to bus 602 for storing information and software instructions to be executed by processor(s) 604. Main memory 606 also may be used for storing temporary variables or other intermediate information during execution of software instructions to be executed by processor(s) 604.

[0079] Software instructions, when stored in storage media accessible to processor(s) 604, render computing device 600 into a special-purpose computing device that is customized to perform the operations specified in the software instructions. The terms “software”, “software instructions”, “computer program”, “computer-executable instructions”, and “processor-executable instructions” are to be broadly construed to cover any machine-readable information, whether or not human-readable, for instructing a computing device to perform specific operations, and including, but not limited to, application software, desktop applications, scripts, binaries, operating systems, device drivers, boot loaders, shells, utilities,

system software, JAVASCRIPT, web pages, web applications, plugins, embedded software, microcode, compilers, debuggers, interpreters, virtual machines, linkers, and text editors.

[0080] Computing device 600 also may include read only memory (ROM) 608 or other static storage device coupled to bus 602 for storing static information and software instructions for processor(s) 604.

[0081] One or more mass storage devices 610 may be coupled to bus 602 for persistently storing information and software instructions on fixed or removable media, such as magnetic, optical, solid-state, magnetic-optical, flash memory, or any other available mass storage technology. The mass storage may be shared on a network, or it may be dedicated mass storage. Typically, at least one of the mass storage devices 610 (e.g., the main hard disk for the device) stores a body of program and data for directing operation of the computing device, including an operating system, user application programs, driver and other support files, as well as other data files of all sorts.

[0082] Computing device 600 may be coupled via bus 602 to display 612, such as a liquid crystal display (LCD) or other electronic visual display, for displaying information to a computer user. In some configurations, a touch sensitive surface incorporating touch detection technology (e.g., resistive, capacitive, etc.) may be overlaid on display 612 to form a touch sensitive display for communicating touch gesture (e.g., finger or stylus) input to processor(s) 604.

[0083] An input device 614, including alphanumeric and other keys, may be coupled to bus 602 for communicating information and command selections to processor 604. In addition to or instead of alphanumeric and other keys, input device 614 may include one or more physical buttons or switches such as, for example, a power (on/off) button, a “home” button, volume control buttons, or the like.

[0084] Another type of user input device may be a cursor control 616, such as a mouse, a trackball, cursor or touch screen direction keys for communicating direction information and command selections to processor 604 and for controlling cursor movement on display 612. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane. Other input device embodiments include an audio or speech recognition input module to recognize audio input such as speech, a visual input device capable of recognizing gestures by a user, and a keyboard.

[0085] While in some configurations, such as the configuration depicted in figure 7, one or more of display 612, input device 614, and cursor control 616 are external components

(i.e., peripheral devices) of computing device 600, some or all of display 612, input device 614, and cursor control 616 are integrated as part of the form factor of computing device 600 in other configurations.

[0086] In addition to or in place of the display 612 any other form of user output device can be used such as an audio output device or a tactile (vibrational) output device.

[0087] Functions of the disclosed systems, methods, and modules may be performed by computing device 600 in response to processor(s) 604 executing one or more programs of software instructions contained in main memory 606. Such software instructions may be read into main memory 606 from another storage medium, such as storage device(s) 610 or a transmission medium. Execution of the software instructions contained in main memory 606 cause processor(s) 604 to perform the functions of the example embodiment(s).

[0088] While functions and operations of the example embodiment(s) may be implemented entirely with software instructions, hard-wired or programmable circuitry of computing device 600 (e.g., an ASIC, a FPGA, or the like) may be used in other embodiments in place of or in combination with software instructions to perform the functions, according to the requirements of the particular implementation at hand.

[0089] The term “storage media” as used herein refers to any non-transitory media that store data and/or software instructions that cause a computing device to operate in a specific fashion. Such storage media may comprise non-volatile media and/or volatile media. Non-volatile media includes, for example, non-volatile random access memory (NVRAM), flash memory, optical disks, magnetic disks, or solid-state drives, such as storage device 610. Volatile media includes dynamic memory, such as main memory 606. Common forms of storage media include, for example, a floppy disk, a flexible disk, hard disk, solid-state drive, magnetic tape, or any other magnetic data storage medium, a CD-ROM, any other optical data storage medium, any physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, NVRAM, flash memory, any other memory chip or cartridge.

[0090] Storage media is distinct from but may be used in conjunction with transmission media. Transmission media participates in transferring information between storage media. For example, transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 602. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications. A machine-readable medium carrying instructions in the form of code can comprise a non-transient storage medium and a transmission medium.

[0091] Various forms of media may be involved in carrying one or more sequences of one or more software instructions to processor(s) 604 for execution. For example, the software instructions may initially be carried on a magnetic disk or solid-state drive of a remote computer. The remote computer can load the software instructions into its dynamic memory and send the software instructions over a telephone line using a modem. A modem local to computing device 600 can receive the data on the telephone line and use an infra-red transmitter to convert the data to an infra-red signal. An infra-red detector can receive the data carried in the infra-red signal and appropriate circuitry can place the data on bus 602. Bus 602 carries the data to main memory 606, from which processor(s) 604 retrieves and executes the software instructions. The software instructions received by main memory 606 may optionally be stored on storage device(s) 610 either before or after execution by processor(s) 604.

[0092] Computing device 600 also may include one or more communication interface(s) 618 coupled to bus 602. A communication interface 618 provides a two-way data communication coupling to a wired or wireless network link 620 that is connected to a local network 622 (e.g., Ethernet network, Wireless Local Area Network, cellular phone network, Bluetooth wireless network, or the like). Communication interface 618 sends and receives electrical, electromagnetic, or optical signals that carry digital data streams representing various types of information. For example, communication interface 618 may be a wired network interface card, a wireless network interface card with an integrated radio antenna, or a modem (e.g., ISDN, DSL, or cable modem).

[0093] Network link(s) 620 typically provide data communication through one or more networks to other data devices. For example, a network link 620 may provide a connection through a local network 622 to a host computer or to data equipment operated by an Internet Service Provider (ISP). ISP 626 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the “Internet”. Local network(s) 622 and Internet use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link(s) 620 and through communication interface(s) 618, which carry the digital data to and from computing device 600, are example forms of transmission media.

[0094] Computing device 600 can send messages and receive data, including program code, through the network(s), network link(s) 620 and communication interface(s) 618. In the Internet example, a server might transmit a requested code for an application program through Internet, ISP, local network(s) 622 and communication interface(s) 618.

[0095] The received code may be executed by processor 604 as it is received, and/or stored in storage device 610, or other non-volatile storage for later execution.

[0096] One aspect provides a carrier medium such as a non-transient storage medium storing code for execution by a processor of a machine to carry out the method, or a transient medium carrying processor executable code for execution by a processor of a machine to carry out the method. Embodiments can be implemented in programmable digital logic that implements computer code. The code can be supplied to the programmable logic, such as a processor or microprocessor, on a carrier medium. One such embodiment of a carrier medium is a transient medium i.e. a signal such as an electrical, electromagnetic, acoustic, magnetic, or optical signal. Another form of carrier medium is a non-transitory storage medium that stores the code, such as a solid-state memory, magnetic media (hard disk drive), or optical media (Compact disc (CD) or digital versatile disc (DVD)).

[0097] It will be readily understood to those skilled in the art that various other changes in the details, material, and arrangements of the parts and method stages which have been described and illustrated in order to explain the nature of the inventive subject matter may be made without departing from the principles and scope of the inventive subject matter as expressed in the subjoined claims.

CLAIMS

1. A method, in a programmable machine, of using a user interface to cause commands to be executed by the machine, the method comprising:

storing shortcut data for a set of one or more users, the shortcut data comprising a plurality of sets of identifiers for shortcuts for a plurality of states of the machine for the set of one or more users, each stored identifier for a shortcut having a corresponding stored machine command, the state of the machine comprising a stage of processing of the machine;

determining a current state of the machine for a said user;

receiving an input from the user;

determining whether the input matches a stored identifier for a shortcut in the shortcut data for the current state of the machine;

executing a stored machine command corresponding to the stored identifier dependent upon the matching to transition the machine to a next state of the machine;

updating stored history data for the set of one or more users; and

updating the shortcut data using the updated history data.

2. A method according to claim 1, including, if the input from the user does not match a stored identifier, determining whether the input is a command, executing the command, updating the stored history data for the set of one or more users, and updating the shortcut data using the updated history data.

3. A method according to claim 1 or claim 2, including outputting the shortcut identifiers to the user for the current state of the machine.

4. A method according to claim 3, wherein the shortcut identifiers are output in an order dependent upon the history data.

5. A method according to any preceding claim, wherein the stored machine command comprises a plurality of sequential commands to be executed by the machine sequentially.

6. A method according to any preceding claim, including synchronizing the stored shortcut data and the stored history data for the set of one or more users with a remote database

of shortcut data and history data for a plurality of sets of one or more users including the set of one or more users.

7. A method according to claim 6, wherein the history data received from the remote database in the synchronization includes user preference data indicating preferences of the set of one or more users, and the shortcut data is updated using the user preference data.

8. A method according to any one of claims 1 to 5, wherein the machine is a multiuser machine used by a plurality of sets of one or more users and the shortcut data is stored for a plurality of sets of one or more users of the machine.

9. A method according to claim 8, including synchronizing the stored shortcut data and the stored history data for the set of one or more users with a remote database of shortcut data and history data for a plurality of sets of one or more users including the set of users of the machine.

10. A method according to claim 9, wherein the history data received from the remote database in the synchronization includes user preference data indicating preferences of the sets of one or more users, and the shortcut data is updated using the user preference data.

11. A method according to any preceding claim, wherein the input from the user includes one or more text characters in a displayed text entry field, the stored identifier for the shortcut includes a plurality of text characters, the corresponding command includes the plurality of text characters, and determining whether the input matches a stored identifier for a shortcut in the shortcut data comprises matching the input one or more text characters with one or more of the plurality of text characters of the stored identifier.

12. A programmable machine providing a user interface for causing the execution of commands by the machine, the machine comprising:

a program memory storing program code; and

a processor for implementing the program code stored in the program memory;

wherein the program code comprises:

code for controlling the processor to carry out the method of any one of claims 1 to 11.

13. A carrier medium carrying processor executable code, which when executed by a processor of a machine, causes the machine to carry out the method of any one of claims 1 to 11.