990292-4
IPOS copy

METHOD FOR ORGANIZING DATA

Field of Invention

This invention relates to a method for organizing data in a computing device and more particularly to a more user-friendly method for organizing personal management information entered as free form text in a handheld personal computer (H/PC).

Background

Electronic organizers such as personal digital assistants (PDA's) and handheld personal computers (H/PC's) have become increasingly popular for personal information management (PIM). Examples of such personal information include information of business contacts, appointments and tasks. These information are usually maintained in separate databases in a H/PC. An example of a H/PC is one running the WindowsCE operating system (OS) available from Microsoft, Redmond, Washington, USA. Each of the databases in the H/PC running WindowsCE is written to and read from through PIM applications such as Contacts, Tasks and Calendar.

The ease and speed with which an electronic organizer can be used to capture and organize personal information is an important consideration factor when a user selects an electronic organizer. Most if not all of the electronic organizers in the market have an integrated capture and organize function. When capturing and organizing data, a user of such an organizer first launches an appropriate PIM application. The launched PIM application then causes a form fill-in dialog (also called fill-in-the-blanks) to be displayed. Such a form fill-in dialog consists of separate fields. A user moves a cursor among the fields to enter and modify data where desired. Each field captures a different piece of data. Data in the various fields are stored as a record in a database, each field in the form fill-in dialog having a corresponding field in the record.

This method of capturing data using a form fill-in dialog is not intuitive to a user who has to quickly record information. The user would usually prefer a two-step approach in which the user captures the information first and organizes them later. Such a capture-first-organize-later approach is more intuitive to the user. In the real world, information is usually received from unscheduled and spontaneous sources, lending this capture-first-organize-later approach to record the information more practical to a user.

The advantage of the capture-first-organize-later approach is best illustrated with a real life scenario. A user is in the middle of a business meeting, and is required to record the following chunk of information which the user is expected to hurriedly record without impeding the progress of the meeting:

> The next status meeting is scheduled for 9 August, at 9-11a.m., in the Einstein Room. Submit a status report to the Operations Manager two days before the meeting, i.e., by 7 August. And for input to the report, contact Mr. John Smith of the Sales and Marketing Department to get the sales forecast for the next quarter. John's phone number is 538-7666, and his email address is john-smith@singnet.com.sg.

To use the form fill-in dialog to record the information on the fly, a user has to quickly assimilate the information to decide what information goes into which PIM application and into which fields of a record in the application. Using the same example as above, a user would first need to break the chunk of information into logical pieces according to the applications that are used to store them. The information when broken down according to the PIM applications is as follows:

1. Status Meeting to be held on 9 August, at 9-11 a.m., in the Einstein Room (to be entered into the Calendar application).

2. Submit Status Report to the Operations Manager by 7 August. Contact John Smith of Sales and Marketing Department for sales forecast for the next quarter (to be entered into the Task application).
3. John's phone number is 538-7666, and his email address is john-smith@singnet.com.sg (to be entered into the Contact application).

From the above example, it is evident that this separation of information is not intuitive to a user who has to quickly record the information. A capture-first-organizer-later is definitely more practical when organizing such information.

In the capture step of the capture-first-organize-later approach, a user can quickly capture spontaneous information on any rough piece of paper like writing pads, sticky note pads, or any piece of paper within easy reach of the user such as newspapers and grocery bills. These pieces of paper which are used for capturing information though practical are apt to be lost. The information is best captured in a more permanent form. And since the information is ultimately organized in the electronic organizer, the electronic organizer is most suitable for capturing the spontaneous information. Applications such as note pads or word processors on the electronic organizers can be used for capturing the data. These captured data can then be organized into the appropriate databases at a later time using the existing form fill-in dialogs provided by the PIM applications.

However, these form fill-in dialogs suffer from disadvantages when used to organize information that is captured in a file on the electronic organizer. The contents of the file that is displayed on the display are usually overlaid by a form fill-in dialog when a PIM application is launched. This overlaying of the form fill-in dialog obstructs the view of the file contents and makes transcribing of information from the file to the form fill-in dialog difficult. The user may have to move the form fill-in dialog to a different location on the display to effectively transcribe information from the file. The display of an

electronic organizer is also small. There is not much room on the display to properly accommodate and manipulate such a form fill-in dialog. In a worst case, the form fill-in dialog may occupy the entire display such that it becomes impossible to view the file. In such a case, the user will have to switch between the displaying of the file and the form fill-in dialog to transcribe the information. Such a method requires steps which include:

- switching to the display of the contents of the file,
- selecting a portion of text in the file,
- activating a copy command to copy the portion of text into a clipboard,
- switching to the form fill-in dialog,
- positioning the cursor in a field in the form fill-in dialog, and
- activating a paste command to paste the portion of text in the clipboard to the field.

To create an entire record requires repeating the above steps for each desired field. This method of transcribing is cumbersome.

The foregoing therefore creates the need for a method of organizing data that lessens or eliminates some of the inconveniences brought about by the above disadvantages.

Summary

The invention provides a method for organizing data which is entered as free form text into separate fields of a record. According to a preferred embodiment, the invention is implemented as an application in a computing device. This application when activated displays the data in a text area on a display of the computing device. The application also provides a group of store-to-field indicia. Each indicium is associated with a field of the record. To store data into a field of the record, a user selects a portion of the data in the text area. The user next actuates an indicium associated with the field of

the record. The actuation of the indicium causes the selected portion of the data to be copied into the field.

Brief Description of Drawings

The invention will be better understood with reference to the drawings, in which:

Figure 1 is a prior art block diagram of a computing device.

Figure 2 is a block diagram of software residing in a memory of the computing device in Figure 1. One of the software is a data capture and organize application software which uses the data organization method in accordance with the present invention.

Figure 3 is a flowchart illustrating the steps of data organization in the data capture and organize application in Figure 2.

Figure 4A is an example of a data capturing dialog of the data capture and organize application used for quick capturing of data into a page.

Figure 4B is an alternative mode of display of the page and a list of descriptions of existing pages created by the data capture and organize application.

Figure 4C is an example of an organize dialog showing text selected from the data capturing dialog in Figure 4A in a text area and fields of a Calendar record for receiving data.

Figure 4D is an example of an organize dialog showing some fields of a Contacts record containing data transferred into them from the text box.

Figure 4E is the same organize dialog as Figure 4D showing more fields of the Contacts record.

Figure 4F is an example of an organize dialog showing text selected from the data capturing dialog in Figure 4A and fields of a Tasks record where data is entered directly into some of the fields in the Tasks record.

Detailed Description of the Preferred Embodiment

A data capture and organize application embodying the invention is described as an application residing in a computing device such as a handheld personal computer (H/PC). Figure 1 is a block diagram of major components of a typical pen-based H/PC 100. The H/PC 100 has a display and input assembly 102, a QWERTY keyboard 104, a memory 106, a pen or stylus 108 and a processor 110. An operating system (OS) which manages the operations within the H/PC 100 resides in the memory 106. The display and input assembly 102 is both an input and an output device. When acting as an output device, the assembly receives data from the processor 110 and displays that data on a suitable display screen. The display screen is preferably a liquid crystal display (LCD). The input device of the display assembly 102 is preferably a thin, clear membrane which covers the LCD display and which is sensitive to the position of the stylus 108 on its surface. With such a structure, the membrane of the display assembly 102 serves as an input tablet.

The data capture and organize application is implemented as a series of program instructions that resides in a program storage device, such as the memory 106 of the H/PC 100. The instructions are collectively known as an application program or simply a program. There are several methods of launching such a word processing application. One of the methods is to have an appropriate application icon appear on the display assembly 102 to provide a means for launching the search application. To launch the application using such a method, a user activates the application icon with the stylus 108. This action causes a control circuit of the display assembly 102 to send a launch application signal to the processor 110. The processor 110 on detecting the signal causes the OS to run the data capture and organize application program.

Figure 2 is a block diagram showing some software that resides in the memory 106 of Figure 1. The memory 106 besides containing the data

capture and organize application program 200 also contains other programs 202, 204, 206, which provide other applications 208, 210, 212 which are separate from the word processing application 213. Examples of such existing applications on a H/PC 100 are a Contacts application 208, a Tasks application 210 and a Calendar application 212. The user activates the Contacts application 208 to store and search for personal information such as names, addresses and telephone numbers of friends and business contacts. The Tasks application 210 is used to maintain a list of outstanding to-dos. The Tasks application 210 holds information such as task descriptions, due dates and others. The user activates the Calendar application 212 to maintain a list of appointments, each of which logs information such as appointment description, date, time and venue.

The applications 208, 210, 212 maintain their respective information in data stores such as files (not shown) or databases 214, 216 and 218 that reside in a readable and writable portion 220 of the memory 106. Each related group of data, for example, a name, an address and a telephone number, is stored in a database as a record. The individual pieces of information in the record, that is, the name, address and telephone number are known as fields. A database contains many instances of each record type. It is important to distinguish between a type and an instance of both a record and a field. For example, an instance of a Contacts record type in the contacts database 218 consists of an instance of each of the following field types: name, address, telephone number and email address. As a further example, the name "John Smith" is an instance of the name field type. It is a common practice to drop the qualifiers "type" and "instance" (for both records and fields), and to rely on context to indicate which of the two is meant. The description that follows adopts this common practice.

To access the databases 214, 216, 218, the applications 210, 212, 213, 214 use services provided by a separate layer of software commonly known as a database management system (DBMS) 224. The DBMS 224 in turn uses services provided by an OS 222 to read and write to the databases 214, 216, 218. The DBMS 224 offers facilities such as facilities for creating

databases, inserting data to and retrieving data from the databases 214, 216 and 218 and other facilities as is known to those skilled in the art.

The operation sequence embodying the data organization method is described with reference to Figure 3. The method is described, as implemented in but not limited to, the data capture and organize application 213. Existing text editing or word processing applications can be used for capturing data. The data organization method according to this invention can be added into these applications for organizing the captured data. The text editing and word processing applications allow text to be entered in a free form manner and do not restrict a user to any particular order of text entry. The applications also do not dictate any particular syntax of text, for example a date entry that must follow a syntax of mm/dd/yy in order to be correctly recognized by the H/PC. Without the need to follow a rigid sequence of text entry allows the user to enter information in any sequence as the information comes to mind or is being conveyed.

Figure 3 is a flowchart showing a sequence of steps 300 embodying the data organization method. When the data capture and organize application 213 is activated, the sequence starts in a START step 302 and proceeds to a DISPLAY DATA CAPTURING DIALOG step 304. In this step 304, a data capturing dialog 400 such as that shown in Figure 4A is displayed on the display 102 of the H/PC. This data capture dialog 400 includes a text box 402 to allow a user to capture information by entering text 404 in the text box 402. The data capture dialog 400 also includes database buttons 406, 408, 410, which when actuated causes data organizing dialogs such as those in Figures 4C, 4D, 4E and 4F to be displayed for organizing captured data into the databases 214, 216, 218.

The data capturing dialog 400 in Figure 4A is now described in more detail. In addition to the text box 402, the data capturing dialog 400 has a current page field 412 which shows a short description or name of an active page in the text box 402. Associated with this current page field 412 is a drop-down button 414. This drop-down button 414 when actuated causes a drop-down menu (not shown) to be displayed. The drop-down menu (not

shown) contains a list of descriptions of pages created with the data capture and organize application 213. To work on any page, the user selects the description of the page on the drop-down menu (not shown).

Alternatively, the application 213 can display the list of descriptions of pages in the data capturing dialog 400 as shown in Figure 4B. The application 213 dedicates a descriptions display area 416 for displaying the list of descriptions of pages. This dedicated descriptions display area allows the user to quickly open a selected page for editing. The user can switch between the two display modes of the data capturing dialog 400 by selecting the appropriate command (not shown) in a View menu 418.

When a page is opened for editing, the sequence proceeds to a first USER INPUT DETECTED? step 306. In this step 306, the application 213 waits for any user entry. When a user entry is detected, the sequence proceeds to a first TEXT EDITING? step 308. If the user entry is a text editing command as determined by the TEXT EDITING? step 308, the sequence proceeds to a MODIFY TEXT step 310, where the text 404 in the text box 402 is updated according the text editing command. Such a method of capturing data allows a user to capture data quickly. The user can save this captured data in the page for future reference or the user may store some or all of the data into the appropriate databases. After the text 404 is updated in the MODIFY TEXT step 310, the sequence returns to the first USER INPUT DETECTED? step 306, where the application 213 waits for further user entry.

If user entry is not a text editing command, as determined in the TEXT EDITING? step 308, the sequence proceeds to a STORE-TO-DATABASE COMMAND? step 312. In this step 312, the application checks if the user entry is a store-to-database command. The actuation of the database buttons 406, 408, 410 will cause such a store-to-database command to be generated. If the user entry is not a store-to-database command, the sequence returns to the USER INPUT DETECTED? step 306, where the application 213 waits for further user entry. There are other user commands which the user can invoke while in the data capturing dialog 400. These other user commands are not

described as they are not pertinent to this embodiment of the present invention.

Prior to the activation of the store-to-database command, the user can select a portion of the text 404 in the text box 402 for organizing. If any portion of the text 404 is selected, only the selected text will be displayed in a subsequent data organizing dialog. If no text is selected, the application will display all of the text 404 in the data organizing dialog. Figure 4C shows a data organizing dialog 420 having a text box 422 which shows all of the text 404 of the earlier displayed data capturing dialog 400.

If the user entry is a store-to-database command as determined in the STORE-TO-DATABASE COMMAND? step 312, the sequence proceeds to a DISPLAY DATA ORGANIZING DIALOG step 314, where the application 213 displays the data organizing dialog 420 on the display 102. In addition to the text box 422, the data organizing dialog 420 has a record fields area 424. The application displays the selected text in the text box 422 and allows the selected text to be edited by the user. The changes in the text in this text box 422 may or may not be reflected in the original text 404. The application displays fields of a record in the record fields area 424 according to the type of the store-to-database command. For example, Figure 4C shows fields of a Calendar record which consists of an appointment description, a location, a date, a start time, an end time and a general notes field. Some of these fields such as the appointment description, location and general notes fields are user editable. Text can be entered into these fields for storing into the Calendar database. To avoid ambiguity in date and time formats in this embodiment, the date, start time and end time fields are not user editable. To enter data into the date field, the user selects a down button to display a calendar from which the user can select a desired date. To enter data into the start time and end time fields, the user highlights the respective hour and minute data and clicks an up and a down button associated with the time field to select the desired time for the appointment.

Figure 4D shows the fields of a Contacts record in the record fields area 424 of the data organizing dialog 420. Such a Contacts record has more

fields than can be displayed simultaneously in the record fields area 424. In such a case, the application 213 provides a scroll bar 426 for the user to scroll through all fields in the Contacts record to cause the appropriate fields to be displayed in the record fields area 424. Figure 4E shows more fields of the Contact record after the scroll bar is moved. Similarly, the application provides two page scroll bars 428, 430 in the text area 422 for bringing different portions of the selected text into view in the text box 422.

After the data organizing dialog 420 is displayed, the sequence proceeds to a second USER INPUT DETECTED? step 316. In this step 316, the application 213 waits for any user entry. If a user entry is detected, the sequence proceeds to a second TEXT EDITING? step 318, where the application checks if the user entry is a text editing command. If it is determined that the user entry is a text editing command, the sequence proceeds to an UPDATE TEXT step 320. In this step 320, the application updates text wherever a cursor is on the display. The cursor may be in the text box 424 or in any of the fields in the record fields area 424. After the UPDATE TEXT step 320, the sequence returns to the USER INPUT DETECTED? step 316.

If the user entry is not a text editing command as determined by the TEXT EDITING? step 318, the sequence proceeds to a TRANSFER BUTTON? step 322. In this step 322, the application checks if the user entry is a transfer-data-into-field command. This command is invoked when any of the transfer buttons 432 is actuated. The transfer buttons 432 preferably appear on the display as a group of indicia adjacent to corresponding fields in the record fields area 424. If it is determined that the user entry is an actuation of a transfer button 432, the sequence proceeds to TRANSFER SELECTED TEXT TO FIELD step 324. In this step 324, the application causes any selected text to be moved or copied into a field associated with the actuated transfer button 432. If no text is selected in the text box 422, no data will be entered in the field. In the event that the selected text spans several lines, the application replaces any non-printable character such as a

"newline" and a "carriage return" character in the selected text with a printable character such as a "space" character in the destination field.

If the user entry is not a transfer-data-to-field command as determined by the TRANSFER BUTTON? step 322, the sequence proceeds to a SEND TO DATABASE? step 326. In this step 326, the application checks if the user entry is a store-data-in-fields-to-database command. The user invokes such a command by selecting a send command (not shown) in a Send menu 434. If it is determined that the user entry is a command to store the data in the fields to a database, the sequence proceeds to a STORE FIELDS DATA TO DATABASE step 328. In this step, the application makes appropriate API calls to store the data in the fields to the database. The sequence finally returns to the DISPLAY DATA CAPTURING DIALOG step 304.

If the user entry is not a command to store data in the fields to a database as determined by the SEND TO DATABASE? step 326, the sequence returns to the second USER INPUT DETECTED? step 314 to wait for a user entry. There are other user commands that can be invoked in the data organizing dialog 420 but they are not pertinent to this embodiment of the present invention and are therefore not described.

Figure 4F shows that text 436, 438 can be entered directly into the fields in the record fields area if the required data is not in the text in the text box 422.

While the present invention has been shown and described with reference to the foregoing operational principles and preferred embodiment, it will be apparent to those skilled in the art that other changes in form and detail may be made.

For example, in the above disclosure, the keyboard 104 is described as a means to enter and edit text in the data capture and organize application. It will be expressly understood by those skilled in the art that a handwriting recognition or voice recognition means is equally applicable for carrying out such tasks of text entry. Also, instead of having a contiguous block of selected text in the data capturing dialog for storing, the selected text may consist of separate blocks of text.

As a further example, the data organizing method need not be implemented in a separate data capture and organize application as described in the above preferred embodiment. The method may be integrally implemented in an application such as a word processing application which allows easy editing of text for capturing data. The implementation may be in the form of a software function within the word processing application.

The data capture and organize application described can be implemented as two separate applications. A first application, such as a word processing application can be used to capture data. A second application when launched, either from within the first application or independently, can present an organize dialog that contains fields and transfer buttons as previously described. The display of the electronic device will be shared by the two applications. The launch of the second application will cause a capture dialog of the first application to displayed in a smaller area of the display to make room for the display of the organize dialog. The procedure for copying text from the capture dialog to the fields in the organize dialog is similar to that described in the preferred embodiment.

In the preferred embodiment, date and time data are not copied from the text box 422 using transfer buttons 432 because of ambiguity caused by the many different date and time formats. To avoid this ambiguity, the application forces the user to choose date and time data by guided user interfaces. However it is possible to include an extra software routine which parses date and time data entered as free form text for translating the text into appropriate date and time formats recognizable by the application.

The transfer buttons may not be necessary if text in the fields is designed to be not editable. Selected text can be copied into such a field by clicking in the field itself. The copied text is then displayed in the field. If the copied text is not required to be displayed, the fields can be totally replaced by a pull-down menu consisting of store-to-specific-field commands or by a series of indicia indicating the different fields of a record. On selecting text in the text box and actuating either a menu command or an indicium, the selected text is copied into a field in the record corresponding to the actuated

menu command or indicium. The transfer buttons can also be implemented as keys on the keyboard.

Claims

I claim:

1. A method of organizing data in a computing device having a display, the data being of free form text for storing into separate fields of a record, the method comprising the steps of:

displaying the data in a text area on the display of the computing device;

providing a plurality of store-to-field indicia each of which when actuated causes selected text in the data to be transferred to a field of the record corresponding to an actuated store-to-field indicium;

receiving a portion of the data as selected text in response to a user's selection; and

copying the selected text into a field of the record in response to the user actuating a corresponding store-to-field indicium.

2. A method of organizing data according to Claim 1, wherein the plurality of store-to-field indicia are displayed on the display of the computing device.

3. A method of organizing data according to Claim 1, wherein the computing device further comprises a keyboard and wherein the plurality of store-to-field indicia are keys on the keyboard.

4. A method of organizing data according to Claim 1, further comprising a step of displaying the contents of the field to indicate the portion of the data which is copied into the field after the copying step.

5. A method of organizing data according to Claim 4, wherein the contents of the field is displayed in an editable field area and the method further comprises a step of modifying the content of the field in response to editing commands from the user.

6. A method of organizing data according to Claim 1, wherein the free form text is selected from a page of captured data.

7. A method of organizing data according to Claim 6, wherein the text area on the display is editable.

8. A method of organizing data according to Claim 7, wherein modifications to the free form text in the text area is reflected in the original page.

9. A method of organizing data according to Claim 1, wherein an unprintable character in the selected text is replaced by a printable character.

10. A method of organizing data according to Claim 9, wherein the printable character is a "space" character.

11. A method of organizing data according to Claim 1, wherein all method steps are implemented in a single software application.

12. A method of organizing data according to Claim 1, further comprising a step of removing the selected text on the display after copying.

13. A program storage device readable by a computing device, tangibly embodying a program of instructions, executable by the computing device to perform the method steps for organizing data in a computing device having a display, the data being of free form text for storing into separate fields of a record, the method comprising the steps of:

displaying the data in a text area on the display of the computing device;

providing a plurality of store-to-field indicia each of which when actuated causes selected text in the data to be transferred to a field of the record corresponding to an actuated store-to-field indicia;

receiving a portion of the data as selected text in response to a user's selection; and

copying the selected text into a field of the record in response to the user actuating a corresponding store-to-field indicia.

* C L M S *