Listpad: Improving Structured Data Entry on Mobile Devices

Kerry Shih-Ping Chang
Human-Computer Inter. Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
kerrychang@cs.cmu.edu

Brad A. Myers
Human-Computer Inter. Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
bam@cs.cmu.edu

Gene M. Cahill
Software Engineering Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
gmcahill@sei.cmu.edu

Soumya Simanta
Software Engineering Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
ssimanta@sei.cmu.edu

Edwin Morris
Software Engineering Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
ejm@sei.cmu.edu

Grace Lewis
Software Engineering Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
glewis@sei.cmu.edu

CHI'13, April 27 – May 2, 2013, Paris, France.
ACM 978-1-XXXX-XXXX-X/XX/XX.

Abstract

Most text entry research focuses on unrestricted text entry. However, a significant amount of use of mobile devices is for personal information management (PIM), where data is highly structured. Listpad is a new application that investigates how the structure inherent in PIM apps can facilitate and even reduce the text entry required. To increase flexibility, Listpad allows users to create custom structures while entering the data and uses data detectors to automate structure definition. Listpad takes advantages of the structure information, along with local and on-line data sources, to provide typing suggestions that are more relevant to what users might enter. A preliminary user study showed that Listpad reduces text entry times by 24%.

Author Keywords

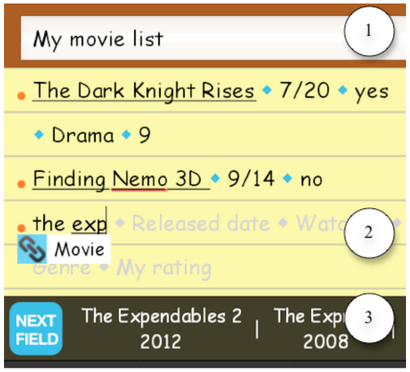
Input, autocomplete, mobile devices, structured data

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces – Interaction styles.

General Terms

Human Factors, Design.



My movie list

- The Dark Knight Rises ◆ 7/20 ◆ yes ◆ Drama ◆ 9
- Finding Nemo 3D ◆ 9/14 ◆ no
- the exp ◆ Released date ◆ Watc ◆ Genre ◆ My rating

Figure 1: The Listpad interface. The user is creating a customized movie list, where (1) is the title box, (2) is the editing interface, and (3) shows Listpad's autocomplete suggestions next to the "Next Field" button.

Introduction

Mobile devices such as smartphones and tablets are becoming the future platforms for personal information management (PIM). Most PIM applications require input data to be in a specific structure, usually by having the users fill in a set of predefined fields. The structure increases the usefulness of the data as it improves readability and is highly utilized to support searching, visualization and data sharing [3]. However, studies have found that mobile device users often reject such PIM applications because the structure is too restrictive and makes entering data cumbersome [2]. Users are forced to follow the predefined fields and data formats, and use special GUI widgets (e.g., calendar, clock) for certain types of input [3]. Although these methods protect the completeness of the data, they also increase users' cognitive load and limit the kinds of data that can be entered [2]. For example, the user cannot enter "around Feb. 20th" in a birthday field in Contacts because the calendar widget does not allow such input. In fact, studies have found that much personal information data does not fit very well into existing PIM applications [1,2]. Also, typing on a touchscreen is known to be slow and error-prone [4], especially for structured data that requires switching between different keyboards. The result of this clumsy data entry process is that much of the information goes uncaptured while people are on the go.

Prior research recommends lightweight and more flexible data entry for mobile PIM applications [1]. Studies show that people often prefer entering data as plaintext in Notepad or Memo because these applications are quick to use and let users enter anything they want [3]. However, unstructured data is less useful, readable and manageable. Conventional mobile database applications allow users to define custom structures to hold the data, but they require users to first set the type and name of each field in the database, and then use entirely different screens to enter the data items, which is tedious. Mobile systems provide word autocomplete using dictionaries and the user's typing history to speed up data entry. However, this is less helpful when entering PIM data such as proper names, addresses or dates.

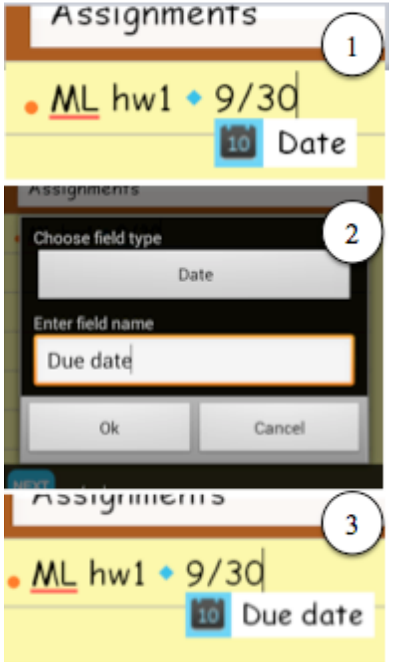
Listpad

We introduce a new mobile PIM application called Listpad (Figure 1), which we implemented for Android smartphones as a research prototype. Listpad contributes novel data entry techniques that allow:

- Easy creation of custom structures while entering data in an unstructured, notepad-like interface.
- Structure-sensitive autocomplete suggestions using relevant local and online databases.

Structure creation and data entry

In Listpad, we define a *list* (similar to a traditional database) as a collection of *items* (similar to a record in a database). Each item has one or more fields. Whereas most lists have a fairly uniform structure, this is not a requirement. Every item in a list could have a different structure. All lists are created and edited in a notepad-like interface (Figure 1). Listpad uses two simple editing rules. First, users type the "enter" key to create a new line and start a new item. Each item starts with an orange bullet, which is automatically inserted when the user enters a new line. Second, users can tap the blue "Next Field" button to insert a blue diamond symbol and start a new field.



Choose field type
Date
Enter field name
Due date

Figure 2: (1) The user is editing an Assignment list. The label under 9/30 shows the field type (the calendar icon) and name (the label text) both being "Date", because Listpad detects the field value (9/30) is in a date format. The user could tap on the label and bring up (2) the dialog box to change the field type and name manually. (3) The label updates after the user changes the field name to "Due date".

As the user types, a small label is displayed under the text to show the type (label icon) and name (label text) of the field being edited. Listpad currently supports 9 different data types: text, number, phone, date, time, address, email, website and link. Except for link (see below), Listpad provides corresponding rule-based data detectors for the other eight data types to try to recognize the type of the current field value as the user types. Once a recognizable format is detected, the field type is automatically changed and the field name is updated to a default name of that type. If the user wants a different field type or name, a tap on the label will bring up a dialog box where they can be changed (see Figure 2).

Every time that the user starts a new item, Listpad suggests structures of existing items in the list using gray text (Figure 1 at 2). The new item will reuse all fields in the selected structure so that the field type and name only needs to be set up once.

Structure-sensitive autocomplete suggestions

An insight that motivates our design is that when users type in a field, they are often giving the system more information about the data than just the text. For example, the data typed in a "location" field in a Calendar is likely to be some place's name. Another insight is that data is often highly inter-related. The data a user types often already exists in some local or online database. For example, the entry in the "location" field is likely to be some place in my city and could be looked up using a web service such as Google Places.

As a proof-of-concept, we implemented four special field types that link to relevant external databases for autocomplete suggestions. These four types (and the associated databases) are Places (Google Places), Movies (Rotten Tomatoes), Albums (Last.fm), and People (the local contact book). The user can set a field to be any of these types from the type dropdown in the dialog box (Figure 2 at 2). Currently, these types are listed as sub-items of a "link" type, suggesting that they are linked to some external sources. When the user starts to type in a linked field, Listpad suggests possible values based on the search results from the linked database (Figure 1 at 3). After the user selects an autocomplete suggestion, Listpad not only inserts the full string to the field being edited, but also provides autocomplete options for other fields of the selected item when the user moves to the next field. For example, in Figure 1 when the user selects a movie names and then goes to the next field, other fields of that movie, such as release date, director and genre that have been retrieved from Rotten Tomatoes, are listed as autocomplete options.

Besides using external databases, Listpad also uses its data detectors to provide typing suggestions. For example, when the user types in a one-digit number, Listpad recognizes that it is a number and guesses that it might be part of a time or date. So it suggests a colon and a slash, and the user does not have to switch to the numeric keyboard to enter these symbols.

Viewing and using the data

Users can choose to view their data in a variety of formats: in a traditional form interface or as a note like in the editor (Figure 1). Listpad takes advantage of having structured data and will provide features such as sorting and searching of items by fields and visualizing items using map and calendar views.

	Average data entry time
Listpad	56.39s
Memento	72.30s (22%)
AK Notepad	76.52s (26%)

Table 1. The average time participants spent entering data in a user study for the three tools, and the time difference slower in percentage with respect to Listpad. Memento and AK Notepad used the default system typing suggestions using a built-in dictionary, while Listpad provided additional suggestions from an external database (in this case, Last.fm). The time differences between Listpad and both Memento and AK Notepad are statistically significant (p<.01). The time difference between Memento and AK Notepad is not significant.

Evaluation

As a preliminary evaluation, we conducted a within-subject study with 15 participants, comparing Listpad with Memento (http://mementodatabase.com/), a mobile database application, and AK Notepad (https://catch.com/resources/mobile/aknotepad/), a notepad application. Participants were asked to enter provided structured data on the mobile device with the three applications. The results (Table 1) show that Listpad's autocomplete helps users cut down their data entry time by an average of 24%.

Conclusions and Future Work

Listpad demonstrates the great potential to speed up structured text entry on mobile devices by suggesting what users might enter using local or online databases that are relevant to the structure. Today, thousands of web services are available for public use. We envision using these data sources to support all sorts of mobile personal data entry. For example, the user could quickly enter a shopping list by having the information autocomplete from Amazon or some grocery web services. To support this, we are developing a software architecture that allows new data sources to be added to Listpad by end users. We have observed that many web APIs have similar structure, and we are designing an interactive tool that allows people to set up connections with such data sources, often without writing complex code. Each data source will be a standalone plugin for Listpad that can be easily installed and shared between users. We will also explore ways to sort the autocomplete options by relevance using context.

Our goal is to make Listpad a general-purpose PIM application that could replace many conventional PIM applications (like Contacts) and further allow a bigger variety of data to be recorded and used more effectively on mobile devices. We feel that Listpad's interaction mechanisms are a starting point on the way to making many different kinds of data entry on mobile devices much more efficient, by using autocomplete to reduce the amount of data that needs to be entered, and by easing the definition and use of structure.

Acknowledgements

This material is based in part upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center (DM-0000139), and in part NSF grant IIS-1116724. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of CMU, the SEI, or the NSF.

References

[1] Bernstein, M., Van Kleek, M., Karger, D., and Schraefel, M.C. Information scraps: How and why information eludes our personal information management tools. *ACM Trans. Inf. Syst. 26*, 4 (2008), 24:1–24:46.

[2] Dai, L., Lutters, W.G., and Bower, C. Why use memo for all?: restructuring mobile applications to support informal note taking. *Proc. CHI*, ACM (2005), 1320–1323.

[3] Van Kleek, M., Bernstein, M., Karger, D.R., and schraefel, mc. Gui --- phooey!: the case for text input. *Proc. UIST*, ACM (2007), 193–202.

[4] Oulasvirta, A., Tamminen, S., Roto, V., and Kuorelahti, J. Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile HCI. *Proc. CHI*, ACM (2005), 919–928