June 04, 2018

A logical layer to interpret user interactions

Alison Lentz

Benny Schlesinger

John Thomas DiMartile III

Gabriel Taubman

Regina O'Dell

A logical layer to interpret user interactions

ABSTRACT

Some computing user interfaces include a large number of user-selectable options which can make it difficult for a user to locate the right option. Another frequent user interface problem is inadvertent or unintended invocation of actions, e.g., by selection of an incorrect icon from two icons placed close to each other. In some user interfaces, simple actions sometimes require a disproportionately tedious sequence of interactions. This disclosure introduces a layer between the user and the application such that user commands or interactions are interpreted in light of past interactions, and corrected, filtered or automated as appropriate. Past interactions are utilized, and interpretation or corrective action is performed only upon permission from the user. For ease of interaction, such permission is obtained, e.g., at initial setup, and is modifiable.

KEYWORDS

- User interface
- User experience
- Accessibility
- Autocorrect
- Predictive UX

BACKGROUND

Users are exposed to a large variety of user interface, as they use devices from different manufacturers that run different operating systems and applications. Some users experience difficulty transitioning between different UI. Some computing user interfaces include a large number of user-selectable options which can make it difficult for a user to locate the right

option. Another frequent user interface problem is inadvertent or unintended invocation of actions, e.g., by selection of an incorrect icon from two icons placed close to each other. In some user interfaces, simple actions sometimes require a disproportionately tedious sequence of interactions.

DESCRIPTION

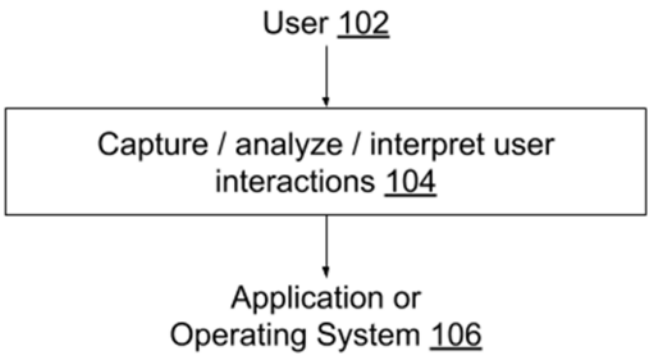

Fig. 1: A logical layer between the user and the application to interpret user interactions

As illustrated in Fig. 1, this disclosure introduces a logical layer (104) between the user (102) and the application or operating system (106). The logical layer is enabled with specific user input. Further, user permission is obtained to analyze past user interactions with the operating system/application and to capture and interpret current user interactions.

The logical layer intercepts or captures user interactions and analyzes such actions with reference to prior interactions. It then interprets the user actions or commands and modifies such commands if necessary. The layer learns from past user actions and patterns to predict if a given user action is to be transmitted to the operating system/application with or without modifications. The layer provides improvement of the UX of an app without a need to change the application itself.

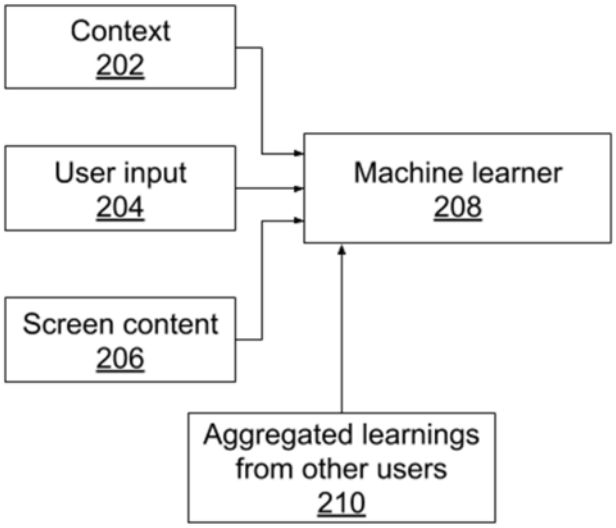

Fig. 2: Learning from past user interactions

Fig. 2 illustrates techniques by which the logical layers learns from permitted user interaction data. With user consent and permission, a machine learner (208) receives contextual information (202), user input (204), and screen content (206). When users consent to use of interaction patterns in determining aggregate patterns across users, the logical layer learns from such aggregated interaction data (210).

The machine learner may be a multi-layer neural network, e.g., a long short-term memory (LSTM) neural network. Other types of models, e.g., recurrent neural networks, convolutional neural networks, support vector machines, random forests, boosted decision trees, etc., can also be used. The machine learner learns from the data, e.g., detects patterns in user interactions with the UI. The predictive accuracy is based on the detected patterns, e.g., by detecting common patterns for retracing of user commands. Examples of detected patterns include: hitting the ESC key to close a dialog box, using cmd-Z/ctrl-Z to undo the previous action, etc.

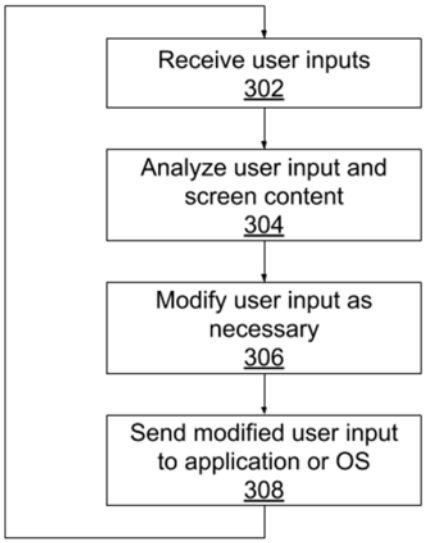

Fig. 3: Using a logical layer to modify user input

Fig. 3 illustrates operation of the logical layer in modifying user input. The logical layer receives user inputs (302). User input includes, e.g., a touch on a touchscreen, a keyboard or mouse event, speech input, gesture input, etc. With user consent and permission, the user input and content of the UI (e.g., displayed on a screen) is analyzed to determine likely user intent (304). Such analysis is based on permitted data regarding prior user actions, patterns, or behavior with reference to the UI.

For example, if an action is often followed by a user undoing the action (e.g., by using cmd-Z/ctrl-Z key combination), the logical layer interprets such a pattern as an indication that that particular action is typically committed inadvertently or in error. Based on the analysis, the logical layer modifies the user input as necessary (306). For example, if a particular action is detected that is frequently undone by the user, then that action is subjected to auto-correction by the logical layer. With user consent and permission, the modified output of the logical layer, known as synthetic user interaction, is sent to the application or operating system (308).

Examples

Example 1 (filtering of erroneous events):

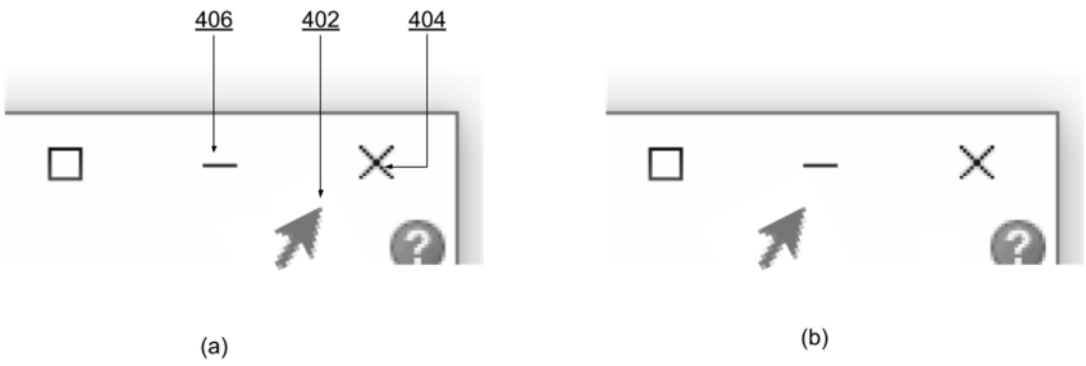

Fig. 4: Filtering erroneous events

Fig. 4 illustrates an example of filtering of erroneous events. In Fig. 4(a), the user intends to minimize a window. However, the mouse pointer (402) is closer to the icon for the close-window command (404) than that for the minimize command (406). Based on previously analyzed interaction patterns, e.g., the closing of windows followed by immediate re-opening by the user, the logical layer interprets the user action of Fig. 4(a) as the minimize command. Fig 4(b) shows the synthetic user interaction generated by the logical layer, which indicates minimization, rather than close-window action.

Example 2 (accessibility): Users that are unable to provide accurate input, e.g., due to a condition such as hand tremors benefit from the techniques described herein. For example, the techniques are applied to perform mouse stabilization and eliminate hand shaking. A user with hand tremor can seamlessly use a mouse or touch screen and permit the logical layer to apply stabilization prior to passing mouse events to the application or operating system.

Example 3 (accessibility): For certain users, e.g., visually impaired users, the logical layer can read out a high-level structure of the UI, rather than a component-by-component read-out of the UI. As in other examples, app developers need not change the application for this user benefit. Similar improvements can be achieved across a range of accessibility tools.

Example 4 (modify input events to fix errors): User data for a user indicates a pattern of the user closing unsent emails, only to immediately re-open them. The logical layer learns to interpret this pattern as unintended action and such user input is ignored.

Example 5 (modify input events to limit user choice): A user overwhelmed by the number of available choices in a user interface may make an unintended selection. To eliminate this problem, e.g., in the case of choice of colors, the logical layer can limit the color palette to specific colors, e.g., to existing colors on the screen. Alternately, a full color palette is offered to the user, but user selection, e.g., via a click event, is translated to the nearest color on the limited-color palette. This is similar to a snap-to-grid function in drawing software.

Example 6 (generation of macros): The creation of a macro that automates a sequence of actions, e.g., for printing with specific format, header, or footer, is a lengthy process that involves many mouse, keyboard, or touch screen events. The techniques of this disclosure automate the generation of macros by determining and applying a frequently-repeated sequence of user actions.

The user can interact with the logical layer described herein using input methods such as speech, keyboard, mouse, touch screen, etc. For example, the user can say “make this grey” to change the color of text or other objects. The user can issue a spoken query to the logical layer such as “where is my flight?” to receive a response. The logical layer uses tone to deduce

intention. For example, a spoken command “give me my flight reservation back!” in loud notes is interpreted as a request of urgency. In this manner, the logical layer uses speech to recognize user intent.

The techniques of this disclosure can be implemented in a variety of ways, e.g., as part of software applications such as word processors, web browsers, operating systems, etc. With user permission, the techniques can also be implemented in hardware, e.g., within a transparent film that is physically attached to a display or input device.

Analysis of user interactions for each specific user is performed by the logical layer with that user’s consent and permission. In situations in which certain implementations discussed herein may collect or use personal information about users (e.g., user data, information about a user’s social network, user's location and time at the location, user's biometric information, user's activities and demographic information), users are provided with one or more opportunities to control whether information is collected, whether the personal information is stored, whether the personal information is used, and how the information is collected about the user, stored and used. That is, the systems and methods discussed herein collect, store and/or use user personal information specifically upon receiving explicit authorization from the relevant users to do so.

For example, a user is provided with control over whether programs or features collect user information about that particular user or other users relevant to the program or feature. Each user for which personal information is to be collected is presented with one or more options to allow control over the information collection relevant to that user, to provide permission or authorization as to whether the information is collected and as to which portions of the information are to be collected. For example, users can be provided with one or more

such control options over a communication network. In addition, certain data may be treated in one or more ways before it is stored or used so that personally identifiable information is removed. As one example, a user's identity may be treated so that no personally identifiable information can be determined. As another example, a user's geographic location may be generalized to a larger region so that the user's particular location cannot be determined.

CONCLUSION

This disclosure introduces a layer between the user and the application such that user commands or interactions are interpreted in light of past interactions (obtained with user permission), and corrected, filtered or automated as appropriate. Past interactions are utilized, and the interpretation. corrective action is performed only upon permission from the user. The techniques of this disclosure provide a more accurate and intuitive user interface.