May 24, 2019

Customizing user interfaces based on usage preferences and practices

Matthew Urquhart

Customizing user interfaces based on usage preferences and practices

ABSTRACT

User interfaces (UI) of applications are designed to provide elements connected to all of the various features regardless of their frequency of use. The inclusion of features rarely used by a user within the application UI can lead to a cluttered interface that wastes screen real estate and makes it difficult for the user to select the elements for frequently used operations, thus slowing the user down. Applications sometimes provide users the option to control the visibility and grouping of the menu options and/or toolbar buttons; however, these options require manual input from the user. This disclosure describes adaptive UI that include and arrange UI features based on recognizing the most frequently used application functions with permission from the user. The UI of the application is then customized for the user based on the most used functions.

KEYWORDS

- User interface (UI)
- Customized UI
- Adaptive UI
- Machine learning model
- Application feature grouping
- Usage-based UI
- Personalized UI

BACKGROUND

Software applications often includes many features that are used rarely while a subset of the features are used by most users on a regular basis. However, user interfaces (UI) of applications are designed to provide elements, such as menu options, buttons, etc., connected to

all of the various features regardless of their frequency of use. The inclusion of features rarely used by the user within the application UI can lead to clutter within the interface, can waste screen real estate, and can make it difficult to select the options for frequently used operations, thus slowing the user down. For example, if a user of a presentation application never uses the feature to jot down speaker notes for a slide, the user must manually choose to hide the notes field for each presentation or live with the reduced screen real estate and potential distraction caused by the visible but empty notes field.

Further, functions within applications are typically grouped according to a set of related tasks, such as editing, file management, etc. These groupings are presented in the form of hierarchical menus and/or toolbars. Applications sometimes provide users the option to control the visibility and grouping of the menu options and/or toolbar buttons, but these options require manual input from the user.

DESCRIPTION

This disclosure describes adaptive user interfaces that include and arrange UI features based on recognizing frequently used application functions with permission from the user. If the user permits, the functions of an application that are used by the user on a regular basis are identified. The UI of the application is customized for the user based on the regularly used functions. If permitted by the user, such custom adaptations can include reorganization of the layout to improve ease of use; removal of unused icons, buttons, functions, fields, etc.; grouping features based on use cases; grouping functions that are frequently used in association with each other; etc.

With user permission, machine learning techniques are utilized to examine the user's use of an application to determine the functions and tools within the application that are used by the

user most often. In addition, a trained machine learning model can identify other relevant aspects, such as the functions and tools that are often used in concert, the layout of the UI preferred by the user, the operational workflow typically employed by the user, etc. The output of the machine learning model is utilized to customize the application UI to fit the user's routine usage practices.

The customization can additionally take into account the application preferences specified explicitly by the user. For instance, when starting to use an application or at a later time, the user may specify their intended use of the application, such as graphic design, photo editing, project management, etc. Many applications provide different standardized UI layouts and features for each of several usage scenarios. While such standardization based on intended use can provide a UI closer to the optimal UI for a user's typical practices, it still requires further adjustment to fit each user's unique needs and practices. The customization involves employing one or more of the custom adaptations described earlier, thus providing a cleaner and better organized UI that can help the user focus and carry out tasks more quickly and effectively.

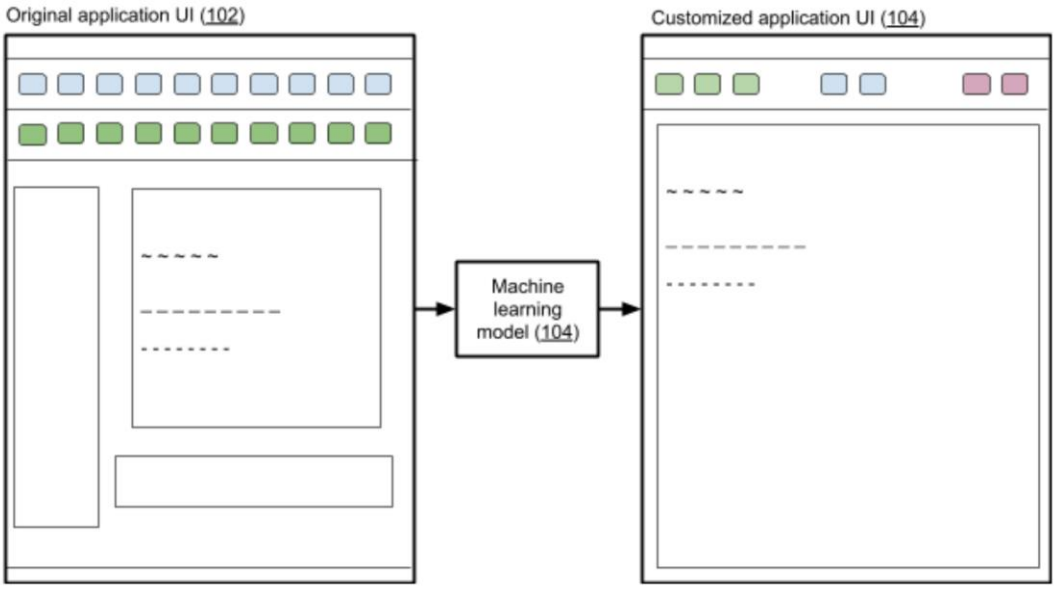

Fig. 1: Machine learning based customization of application UI

Fig. 1 shows an example application of the techniques of this disclosure. A user is presented with original UI of an application (102) upon starting the use the application. The original UI provides access to the various functions of the application via a number of toolbars, buttons, fields, menus, etc. With the user's permission, the user's use of the application is analyzed by a machine learning model (104) to determine the application features used most frequently by the user and/or to identify features that are not used by the user. Based on this determination, the output of the model is employed to generate a customized application UI (104) that retains only those features that are suitable for the user's typical usage requirements. As a result, the customization reduces clutter within the UI and provides more screen real estate for serving the user's priorities.

Changes in UI that hide, reorganize, or relocate functions or features have the potential to create confusion and frustration. Therefore, the user is provided with appropriate controls to determine when and how the UI adaptations proposed in this disclosure are activated. Moreover, the user can override the UI adaptations proposed by the machine learning model. The UI adaptations approved by the user, which include any manual adjustments made to the adaptations originally proposed by the machine learning model, can be saved as the new default UI, thus allowing the adjustments to persist across application sessions. At any time, the user can toggle between the customized UI and the standard out-of-the-box UI of the application.

Any suitable machine learning algorithm can be employed to implement the described techniques. Moreover, the techniques described in this disclosure can be applied to any software that provides a user interface for user interaction. Operational examples of the application of the techniques with user permission include:

- Visibility of labels for emails or files can be determined based on frequent searches.

- Speaker notes for presentation slides can be hidden if the user never adds contents to this field.
- Photos can be grouped according to sharing, searching, or viewing practices.
- Commonly used formatting options for a spreadsheet or word processor can be recognized and grouped together such that they can be applied with a single button click.
- Calendaring applications can make suggestions for meeting times based on the user's preferred meeting blocks.
- Shortcuts for applications or files on user devices, such as smartphones, tablets, computers, etc. can be grouped and organized by usage frequency and/or function.

Use of the described techniques enables the user to benefit from an optimized UI while maintaining control over the interface. The optimized UI can improve the layout and organization of UI elements and features, thus reducing the time required to perform frequent tasks and enhancing the user experience.

If users permit, application developers can use the information about frequently used functions of an application to determine and prioritize the design and development of features in future versions of the application.

While the foregoing description refers to software applications, the described techniques can also be applied to operating systems. For example, the described techniques can be utilized to determine the organization of applications on the home screen of an operating system, for selective inclusion of applications in a quick-access area, etc.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user's social network, social

actions or activities, profession, a user's preferences, or a user's current location), and if the user is sent content or communications from a server. 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. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

This disclosure describes adaptive UI that include and arrange UI features based on recognizing the most frequently used application functions with permission from the user. If the user permits, machine learning techniques are utilized to analyze the user's use of an application to determine the functions and tools within the application that are used by the user most often. In addition, a machine learning model can identify other relevant aspects, such as the functions and tools that are often used in concert, the layout of the UI preferred by the user, the operational workflow typically employed by the user, etc. The output of the machine learning model is applied with user permission to customize the application UI to fit the user's routine usage practices. The techniques described in this disclosure can be applied to any software that involves user interaction. Application of the described techniques enables users to benefit from an optimized UI while maintaining control over the interface. The optimized UI can improve the layout and organization of UI elements and features, and reduce the time required for the user to perform frequent tasks, thus enhancing the user experience.