

Technical Disclosure Commons

Defensive Publications Series

October 03, 2019

Multilingual User Interface For Collaborative Software

Steffen Meschkat

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Meschkat, Steffen, "Multilingual User Interface For Collaborative Software", Technical Disclosure Commons, (October 03, 2019)
https://www.tdcommons.org/dpubs_series/2544



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Multilingual User Interface For Collaborative Software

ABSTRACT

Many operating systems and other software applications include features that enable users to customize the language used for text elements in the user interface. At runtime, the software renders the UI using text elements corresponding to the language chosen by the user or a default language. In certain situations, e.g., for collaborative software that multiple users use simultaneously, this can be less than ideal, e.g., when the different users have different language preferences. This disclosure utilizes a suitable form of UI composition that makes UI text elements available in multiple languages at the same time. With user permission, user preferences of the multiple users as well as the area of UI that individuals users focus on are used to update the composition dynamically. A flexible layout is provided such that the user interface framework can handle different languages, including different sizes of labels.

KEYWORDS

- Multilingual User Interface (UI)
- Collaboration software
- Multi-user software
- Electronic whiteboard
- Conferencing system

BACKGROUND

Many operating systems and other software applications include features that enable users to customize the language used for text elements in the user interface. Text elements for menu options, buttons, warnings, error messages, etc. are translated to all languages supported by the system as a part of the internationalization process (also known as i18n). The generated set of

text in all supported languages is then included in the software or OS package. At runtime, the software or OS renders the UI using the text elements corresponding to the language chosen by the user. If the user has not made an explicit selection for the language of the UI text, then the default language is used. In certain situations, e.g., for collaborative software that multiple users use simultaneously, this can be less than ideal, e.g., when the different users have different language preferences.

In those situations, users can find it beneficial to have the text elements of the UI displayed in more than one language at the same time. Such software may be implemented on systems such as a shared electronic whiteboard in a conference room, a video conferencing system used for multi-person meetings, a television or other entertainment device that includes a content browsing and/or recommendation system suggests content for multiple co-present users, etc.

Currently, multi-user scenarios such as the above are supported using one or more of the following approaches: having the multiple users choose one from among the various languages they prefer; defer to the default language configuration for the device on which the software runs; choosing a primary or active user and their language preference; etc.

DESCRIPTION

This disclosure describes techniques that can be implemented as part of an operating system or other software to make the UI text elements of a system available to the user in multiple languages at the same time. To this end, a user can select a set of one or more languages from those available for displaying the UI rather than choose a single language as is current practice.

The software is implemented such that various UI widgets, such as menus, dialogs, etc. are extended appropriately to have the capability to support multiple languages of the UI text being active at the same time. Specifically, the capabilities of the UI elements in the UI framework used for developing the software are enhanced. The enhancements allow these elements to access text corresponding to the multiple languages supported by the software package and, in turn, to accommodate simultaneous display of the UI text in more than one language.

Situations involving multiple users are supported by displaying the UI text using the set of languages formed by combining the UI language preferences of the users, obtained with their permission. The techniques described herein support displaying the UI text in multiple languages via additional mechanisms that are suitable for cases involving multiple users.

For instance, with user permission, the language of the UI text can be changed dynamically based on contextual aspects, such as user presence or focus of attention. If particular users, e.g., English-readers, focus on different areas of the UI than other users, e.g., German-readers, the corresponding areas can be utilized to display UI text elements in the corresponding language. Further, user presence can be detected, with user permission, and a set of languages to be used for the UI can be determined, e.g., a minimum set of languages determined based on the preferences of the users present. For example, when multiple users watch a television screen together, the UI language can be chosen based on who is present.

To implement multilingual UI, the meaning of the term UI element in the UI framework is updated to enable multiple text labels. The UI element is then connected to a translation database to retrieve a label in the set of languages, e.g., determined based on the users that are present.

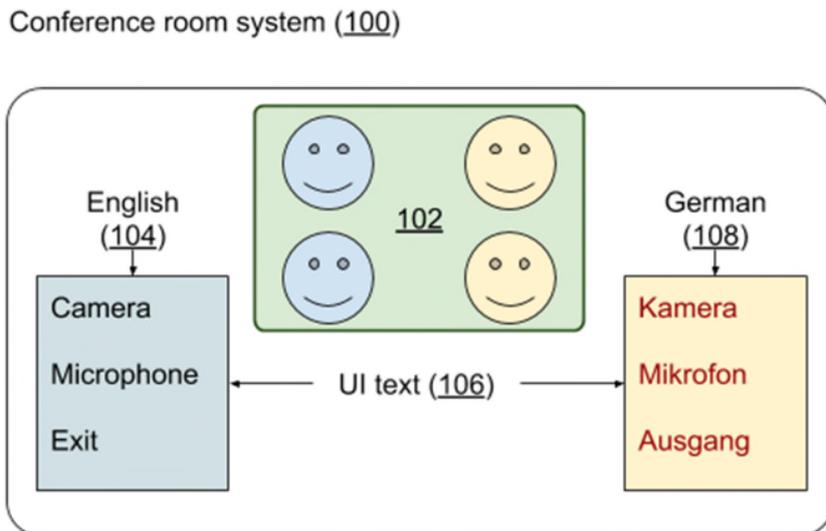


Fig. 1: Multilingual user interface for collaborative software

Fig. 1 shows an example multilingual user interface of collaborative software. A conference room system (100) runs the collaborative software to provide a video conference (102). Four local participants participate, as illustrated in Fig. 1. Two of the participants (blue) are identified as having a preference for English, while two other participants (yellow) are identified as having a preference for German. Accordingly, the conference room system provides a menu of options for the conference in English (104) and German (108). The menus have UI text (106) that enables each participant to read the menu options.

While Fig. 1 illustrates simultaneous display of all UI options, the actual operation of the UI can be selected by the developer of the software and/or based on user preference. For example, if users permit, gaze detection can be performed to identify users that are gazing at particular areas of the display and corresponding UI elements can be shown in the users' preferred language. Alternatively, each of the active UI languages may be associated with one or more parameters, such as a display order, priority, rendering mode, cycle time, etc. which are

used when rendering the UI. Moreover, the UI rendering can vary based on the number of languages chosen for the UI.

The techniques described in this disclosure obviate the need for users to choose whether to keep switching between UI languages or accept the limitations of working with UI text shown only in a single language. As such, implementation of these techniques can enhance the user experience of a variety of multi-user scenarios in which it is beneficial to have the UI text available in multiple languages at the same time. Moreover, implementation of the techniques can result in enhanced support for internationalization, thus contributing to making systems broadly accessible.

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 utilizes a suitable form of UI composition that makes UI text elements available in multiple languages at the same time. With user permission, user preferences of the

multiple users as well as the area of UI that individuals users focus on are used to update the composition dynamically. A flexible layout is provided such that the user interface framework can handle different languages, including different sizes of labels.