

Technical Disclosure Commons

Defensive Publications Series

November 2019

Run time translation of user interface text via an operating system translation service

Steffen Meschkat

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

Recommended Citation

Meschkat, Steffen, "Run time translation of user interface text via an operating system translation service", Technical Disclosure Commons, (November 06, 2019)

https://www.tdcommons.org/dpubs_series/2658



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.

Run time translation of user interface text via an operating system translation service

ABSTRACT

User interface text labels are translated into multiple languages to support users who speak the corresponding languages. Software developers use translation services to generate translations of each label in the UI in multiple languages. The translated labels are packaged with the application when it is built. This approach provides a less than satisfactory user experience since the same text label may be translated differently in applications from different vendors. Further, user preferences are not taken into account when generating translated text labels. The techniques described in this disclosure generate translated labels by taking into account user preferences and serve to avoid inconsistent translation of UI labels across applications. A translation service provided by the operating system is invoked and translates the UI text labels at run time instead of at build time.

KEYWORDS

- Text labels
- User interface (UI)
- Machine translation
- Application package
- Internationalization
- Translation service
- Multiple language support
- Multilingual UI
- UI localization

BACKGROUND

The user interface (UI) of an application usually contains text that labels various elements within the UI. To internationalize the application, such labels are translated into multiple languages in order to support users who speak the corresponding languages. Software application development frameworks that deal with UI aspects typically include mechanisms to manage labels for UI elements in multiple languages. Usually, these mechanisms are part of the processes invoked during building and packaging of the application. As a result, such incorporation of the translated text within the UI occurs when the application is created (at build time), not at the instant when the user invokes and executes the application (at run time).

To incorporate translated text, the corresponding label is included in the argument of a specific function dedicated to the purpose. When building an application, these function calls are detected and a catalog of their arguments is created. A translation service is then used to generate translations of each of the labels in the catalog in multiple languages. The translation service can use manual and/or automated approaches for translating a given piece of text from the original language to the specified language(s).

The translated labels are packaged with the application when it is built. When the application is executed, the function calls are invoked with the original argument. The function implementation is designed to look up the label translations bundled with the application and use the translation of the argument in the language selected by the user.

The above approach suffers from two problems that lower the quality of the translated labels, consequently having a negative effect on the quality of the user experience (UX). First, the same text label that appears in the UIs of different applications from different vendors may be translated differently in each of the applications depending on differences in the translation

services utilized by the respective vendors. The degradation of UX resulting from such inconsistent labeling across applications is particularly noticeable in platforms that promote a tightly integrated UX across applications. Second, user preferences are not taken into account when generating translated text labels. For instance, in languages that involve variations in addressing a person based on formality, the same label can be translated in multiple ways depending on the level of formality. However, the current translations result in only a single variation being generated and included with the application.

DESCRIPTION

The techniques described in this disclosure generate translated labels by taking into account user preferences and serve to avoid inconsistent translation of UI labels across applications. To that end, the techniques involve invoking the translation service for translating the UI text labels at run time instead of at build time, as is done currently.

The approach requires no change in how the application code is written. Further, the building and packaging processes can stay the same, including the inclusion of a catalog of UI text labels extracted from the relevant function calls and corresponding translations obtained from a translation service. However, when the pertinent functions are executed at run time, a translation service provided by the underlying operating system (OS) is first invoked instead of using the translated label packaged with the application.

When invoked, the translation service of the OS can operate in any of the following ways or any combination thereof:

1. Return the translation corresponding to the label that is included with the application, thus operating in the same way as current operations.

2. Return a translation corresponding to the label from a global catalog of label translations that contains text translations reconciled across all applications present on the platform.
3. Return a translation corresponding to the label from an automated machine translation service running locally on the device.
4. Return a translation corresponding to the label from an automated machine translation service external to the device, when the user permits such operation.

In the last two of the above mechanisms, the machine translation can be generic across all users of a given language. Alternatively, or in addition, if the users permit, the translated labels can be personalized by taking into account user preferences and/or translation history. With user permission, the system can record user-specified overrides for any of the automatically translated text labels. This can be provided as a user-editable preference setting.

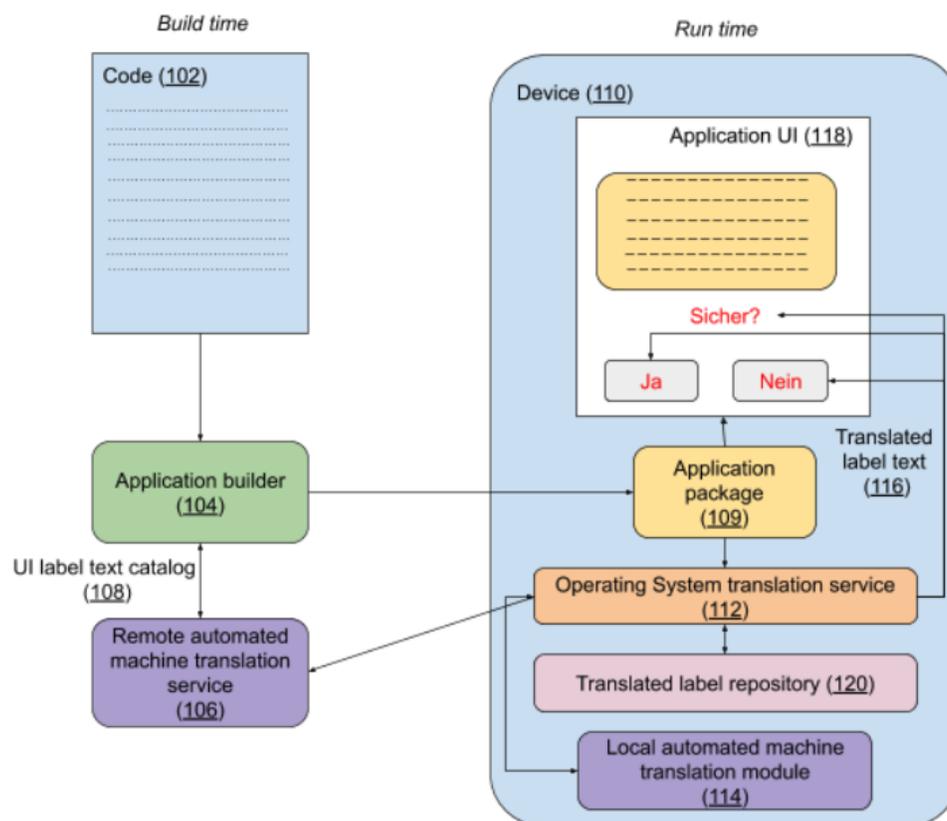


Fig. 1: Run time translation of UI text labels

Fig. 1 illustrates an example implementation of the techniques described in this disclosure. At build time, application code (102) is processed by an application builder module (104). During the build process, a catalog of UI text labels (108) is created and corresponding translations in various languages are obtained from an automated machine translation service (106). The translations are included in the built application package (109).

At run time, the application package is executed on a user device (110). During execution, a translation service provided by the device OS (112) is invoked as needed for translating text elements within the application UI (118). The OS translation service can obtain the needed translations from the application package, from a repository of translations for all applications on the OS (120), a local automated machine translation module on the device (114), the remote automated machine translation service (106), or any combination thereof.

The translated label text (116) returned by the OS translation service is then utilized to display text labels within the application UI. As shown in Fig. 1, original English text within the UI is translated into German (shown in red) at run time by applying the approach described in this disclosure. Further, with user permission, the run time translation is performed according to the user's preferences for being addressed in an informal manner.

The proposed approach leverages the increasingly common use of automated services for generic as well as personalized translation. As the quality of the services improves, they are likely to be highly effective in producing high quality translations of UI text labels.

Implementation of the approach described above requires that the function dedicated to marking UI text as translatable be capable of asynchronous execution. Such asynchronous execution is not yet supported in traditional systems, e.g., systems in which translations are obtained via synchronous calls within the same application rather than asynchronous calls from

the OS. However, applications written using asynchronous code are increasingly common, thus making it feasible to change the relevant application framework such that translated text can be obtained via asynchronous function execution.

Further to the descriptions above, a user is 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 preferences, such as preferred languages and/or level of formality for a language, a user's current location, etc.), and if the user is sent content or communications from a server, e.g., translations of UI text labels. 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, with user permission, prior user selections of language and/or formality level may be accessed to enable translations of UI text labels. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user. Thus, the user has control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

The techniques described in this disclosure generate translated labels by taking into account user preferences and serve to avoid inconsistent translation of UI labels across applications. A translation service provided by the operating system is invoked and translates the UI text labels at run time instead of at build time. The approach requires no change in the processes of coding, building and packaging a software application. With user permission, the translated labels can be personalized based on user preferences.