Patterns for UI language-switching in applications designed for multilinguals

DIEGO MOREIRA DA ROSA, Pontifical Catholic University of Rio Grande do Sul, Brazil and Federal Institute of Rio Grande do Sul, Brazil
YASMIN SILVA NUNES, Pontifical Catholic University of Rio Grande do Sul, Brazil
MILENE SELBACH SILVEIRA, Pontifical Catholic University of Rio Grande do Sul, Brazil

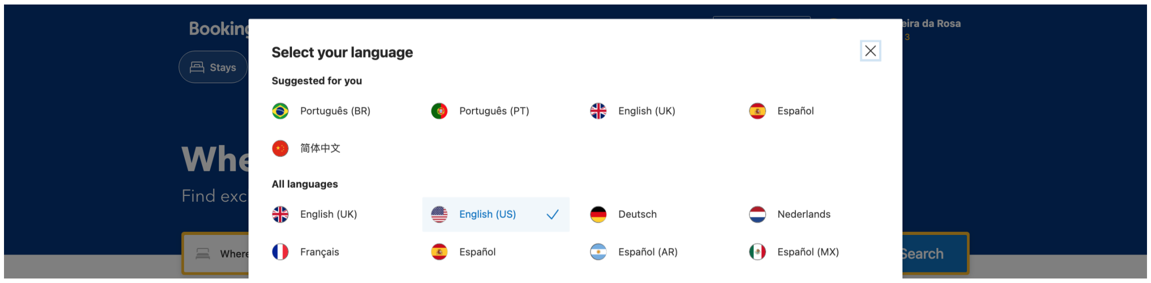


Fig. 1. Homepage of Booking.com (September 2022): users can change the interface language at any time during the interaction.

Current software solutions already present several adaptations regarding the interaction with bilingual and multilingual users, including the presentation of user generated content in multiple languages. In a previous study, through the analysis of existing systems, a language of 30 interaction patterns was proposed, documenting the design strategies present in modern applications aimed at multilinguals. After a first assessment by means of shepherding and a writers' workshop in the European PLoP conference, the language was revised and one new pattern was included. In this paper, we present a renewed version of the language now named Design4Multilinguals, including the new pattern UI LANGUAGE-SWITCHING and six patterns remodeled from the first versions present in the original draft. All patterns of the subset presented here serve the purpose of structuring applications designed for multilinguals or, more specifically, allowing the users to choose and modify at any moment their preferred languages.

CCS Concepts: • **Human-centered computing → Human computer interaction (HCI)**; *Interaction design process and methods.*

Additional Key Words and Phrases: bilingual, multilingual, interaction patterns, pattern language

ACM Reference Format:
Diego Moreira da Rosa, Yasmin Silva Nunes, and Milene Selbach Silveira. 2022. Patterns for UI language-switching in applications designed for multilinguals. In *PLoP '22: Pattern Languages of Programs Conference, October 18, 2022, online.* ACM, New York, NY, USA, 17 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Current software designers are gradually adapting user interfaces to an increasing audience of bilingual and multilingual users. These adaptations include, for example, allowing users to choose a preferred list of languages or presenting user generated content in multiple languages. This movement is in tune with studies from the area of Linguistics which indicate that there are more bilingual and multilingual individuals in the world than there are monolinguals [10]. Wei goes further and affirms that, besides the fact that one in three of the world's population routinely uses two or more languages in everyday life, there are even more people who make irregular use of languages other than their native one: "if we count these people as bilinguals, then monolingual speakers will be a tiny minority in the world today" [11].

Linguists also agree that bilinguals are frequently ignored or oppressed by a monolingual elite. According to Kaplan and Baldauf, for example, it is more often than not the case that a given arrangement of languages benefits only those who have influence and privileges [8]. Romaine, in turn, states that although monolinguals are a minority when considering the world as a whole, they are a very powerful minority, often imposing their language on others, who have no choice but to become bilingual [9]. Despite the estimate of more than 7,000 languages spoken in the world nowadays [6], fewer than 30% of the world's approximately 200 countries recognize two or more official languages, and less than 10% recognize more than two[1]. Hence, designing with multilinguals in mind may represent more than only targeting a large consumer public of mobile apps and online services. It could be a key measure for the digital inclusion of a large portion of the world's population, helping to guarantee the survival of many endangered languages.

In 2014, intrigued by the fact that Google insisted on hiding multilingual content in their Play Store mobile app[2], Hale remarked in a blog post the importance of *designing for multilinguals* [7]. Despite the colloquial writing of the text, the author gives two indications on how to achieve that goal:

(1) Allow each user to have a set of multiple preferred languages;
(2) Consider bilingual and multilingual users when designing platforms.

Although the second recommendation may seem broad and subjective, the first one is more objective. Instead of determining only one language for the interface based on the user's locale or IP address, systems should allow users to define a set of preferred languages and, consequently, should allow them to change this list as they wish. Possible drawbacks of this adaptation, such as reduced usability or increased cost/complexity, must also be considered and this strategy may not be the best solution for all types of software. For example, systems with few user generated content or with only a regional distribution would probably benefit little from this type of design. Nonetheless, designing for multilinguals fit well to a great number of Web 2.0 and mobile applications that allow plenty of user interaction and that are internationally distributed. Currently, many websites and applications already present the adaptations predicted by Hale. That is the case, for example, of the accommodation website Booking.com[3], which displays user reviews written in many different languages and allows its users to change the language and currency at any time during the interaction (see Fig. 1). In contrast, both Play Store and Apple's App Store[4], still hide user reviews written in languages other than the configured interface language.

In a previous study, through the analysis of existing systems, we proposed a language of 30 interaction patterns, documenting the design strategies present in modern applications aimed at multilinguals [4]. This study was accepted for publication in the European Conference on Pattern Languages of Programs[5] (EuroPLoP 2022) and was assessed through the traditional processes of shepherding and writers' workshop. Besides a general description of the language, the final version of the paper presented at the workshop also contained the detailed descriptions of three patterns:

[1]Numbers based on Tucker (1998) [10], Ethnologue [6], Worldometer (https://www.worldometers.info), Infoplease (https://www.infoplease.com), and other online sources.
[2]https://play.google.com
[3]https://www.booking.com/
[4]https://www.apple.com/app-store/
[5]https://europlop.net/

MULTILINGUAL OUTPUT, NATIVE LANGUAGE FIRST, and RECOMMENDED CONTENT SORTER. Initial drafts of the remaining 27 patterns were made available through a website, but were not fully described on the paper and were not discussed with the same level of details during the workshop.

From the feedback and insights obtained in the conference, the language was revised and one new pattern was included. In this paper we present the renewed version of the language now called Design4Multilinguals, including the new pattern UI LANGUAGE-SWITCHING and six remodeled patterns. It is worth noting that these six patterns were not fully described in the original paper that introduced the language. While the original paper focused more on multilingual output, the patterns presented here all serve the purpose of language-switching in applications designed for multilinguals or, more specifically, allowing users to choose and modify at any moment their preferred languages.

This paper comprises four sections including this introduction. Section 2 presents an overview of the previous studies and the methodology applied during the patterns discovery process. The seven patterns are described in details on Section 3. Final considerations and future work enclose the paper in Section 4.

2 PREVIOUS WORKS AND METHODOLOGY

In an attempt to assess systems' adaptations to multilinguals, we conducted two studies applying inspection methods based on semiotic engineering[6] to evaluate the presentation of user reviews in current websites [2, 3]. One of these studies also contained a small set of 3 proto-patterns aimed at the design for multilinguals [2]. Through exploratory research, we expanded the list of inspected systems to a total of 33, including operating systems, desktop applications, mobile apps, and websites. The analysis of these systems resulted in a pattern language comprising 30 patterns (see Fig. 2).

The discovery process of the patterns followed best practices from the Software Engineering and Human-Computer Interaction (HCI) pattern communities. A planning phase consisted of performing a domain study, an exploratory research, and an HCI patterns literature review. For the pattern mining, we performed patterns identification, writing, rating, and organization. As stated previously, a paper describing the language in general and detailing three of its patterns was approved for EuroPLoP 2022 and underwent the regular phases of shepherding and writers' workshop. In this paper, we intend to keep the focus on another subset of the patterns (related to user interface language-switching), intentionally omitting a more detailed description of the methodology. We present one new pattern and six rewritten patterns, none of which had previously been described in details. More information on the methodology and discovery procedures can be found on the original paper describing the first draft of the pattern language (to be published) [4].

3 THE PATTERNS

In this paper we detail a subset of the pattern language proposed in our previous works. A website has been constructed and can be accessed for detailed descriptions of all patterns[7]. After the assessment of the language at EuroPLoP, one new pattern was included (UI LANGUAGE-SWITCHING) and two previously described patterns were merged with others (COMPACT INTERFACE LANGUAGE SELECTOR and COMPACT CURRENCY SELECTOR), totaling the 29 patterns presented here.

As part of the organization of the language we classified the patterns according to their abstraction level (marked by the gray horizontal strips in Fig. 2). Patterns termed as *high-level* address large-scale design issues, such as: interface configuration (UI LANGUAGE-SWITCHING), information output (MULTILINGUAL OUTPUT), information input (MULTILINGUAL INPUT),

[6]Semiotic Engineering is a semiotic theory of Human-Computer Interaction (HCI) originally proposed by Clarisse De Souza[5]. It views HCI as a computer-mediated communication between designers and users at interaction time.
[7]https://design4multilinguals.wixsite.com/website

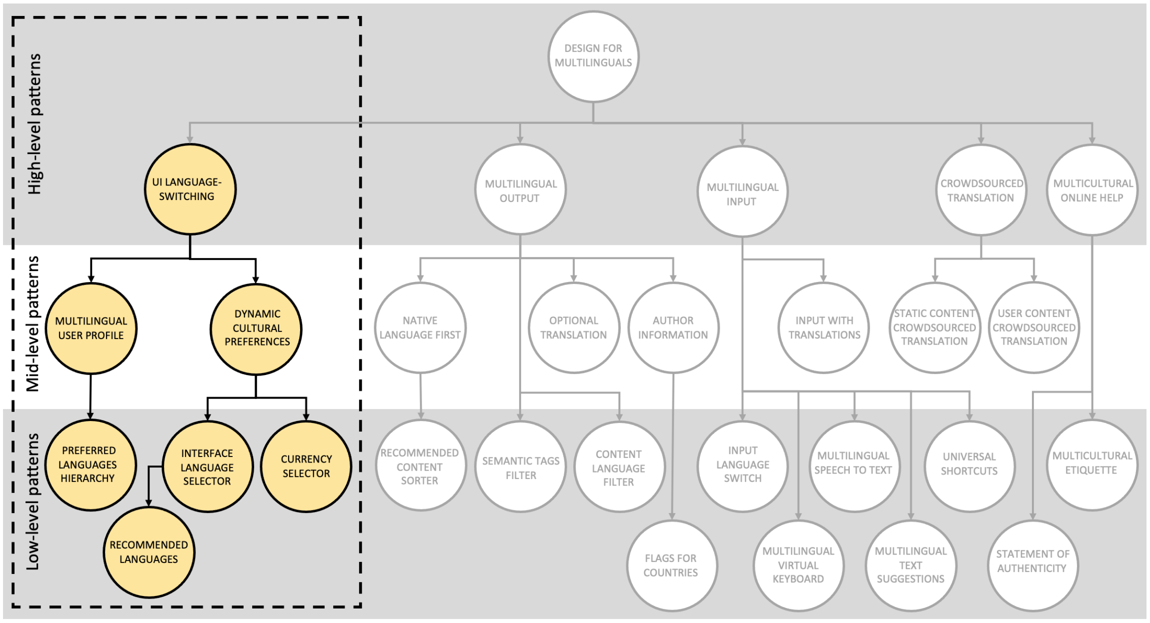


Fig. 2. Complete diagram of the pattern language highlighting the patterns discussed in this paper.

software translation/localization (CROWDSOURCED TRANSLATION), and online help (MULTICULTURAL ONLINE HELP). Patterns dealing with small design details, usually interface widgets, were classified as *low-level* patterns. Finally, patterns with an intermediate abstraction level, were classified as *mid-level* patterns. Fig. 2 shows an overview of the pattern language with the seven patterns detailed here highlighted inside the dashed rectangle:

(1) UI LANGUAGE-SWITCHING: This is the only pattern that was not present in previous descriptions of the language. It describes how current applications provide one single version of the system suited for users of all languages.
(2) MULTILINGUAL USER PROFILE: User profiles adapted for the characteristics of multilingual users.
(3) PREFERRED LANGUAGES HIERARCHY: A list of preferred languages configured by multilingual users.
(4) DYNAMIC CULTURAL PREFERENCES: Controls that allow users to change cultural preferences at interaction time.
(5) INTERFACE LANGUAGE SELECTOR: A control to select one language/locale from a list. This pattern was merged with COMPACT INTERFACE LANGUAGE SELECTOR for the sake of simplicity.
(6) RECOMMENDED LANGUAGES: The recommendation of languages based on collected statistics or previous user activity.
(7) CURRENCY SELECTOR: A control to select one currency from a list. This pattern was merged with COMPACT CURRENCY SELECTOR for the sake of simplicity.

Next, each of the seven patterns are presented in a format inspired by the work of Borchers [1] and following typical PLoP conferences guidelines for pattern structure. All patterns described here can be found in a large number of the inspected systems, hence they are considered of high confidence.

3.1 UI LANGUAGE-SWITCHING

A single version of the website or application is suited for all users (regardless of language and country of origin), who can choose their preferred interface language at interaction time. For an example of this pattern as implemented by Booking.com, see Fig. 1 on the beginning of this paper.

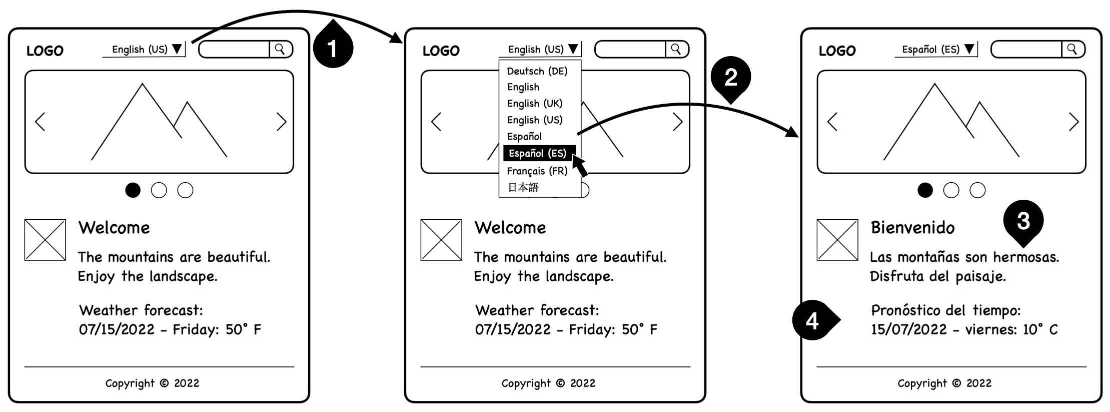
3.1.1 Context. The developer needs to find the best strategy for language selection in a website/application based on DESIGN FOR MULTILINGUALS.

3.1.2 Problem. Multilingual users want to switch the interface language quickly during the interaction without having to access a new URL or install a new application.

3.1.3 Forces.

- When browsing the Web, users don't want to navigate to a new URL to access the website for some specific locale.
- In desktop and mobile applications, users don't want to have to choose a specific locale at installation time.
- When changing the interface translation/localization, users expect minor or no changes in the general look and feel of the interface.
- Users do not want interruptions in their workflow when changes to the interface are applied.

3.1.4 Solution. **A single version of the application/website is provided for users of all the languages/locales in which the application is localized into and users can choose their preferred interface language at interaction time.** A list of available languages/locales is provided to the user. Typically the same source code and infrastructure is shared by all versions of the application but special adaptations to the interface can be made at runtime to adapt the system to different locales. These adaptations are usually subtle and keep the overall look and feel of the user interface. Fig. 3 shows a diagram for the pattern.



1. The application provides mechanisms for the user to change the interface language/locale at interaction time
2. The locales in which the application has been localized determine the available options to the user
3. All text strings are translated accordingly
4. Cultural variables such as date formats and units of measure are also adapted

Fig. 3. Diagram for the UI LANGUAGE-SWITCHING pattern.

3.1.5 *Implementation.*

The initial language/locale of the interface can be determined in different ways: by the configured locale of the user's device; by the history of user activity on the application/website; or by the IP address of the current connection. Two main strategies can be used for interface language selection:

(1) Users select their preferred language by accessing and modifying a MULTILINGUAL USER PROFILE;
(2) Users select DYNAMIC CULTURAL PREFERENCES at anytime during the interaction by means of an INTERFACE LANGUAGE SELECTOR present on the application pages/screens.

The implementation details of UI LANGUAGE-SWITCHING depend on the environment of the system.

Web applications: Many web applications nowadays adopt UI LANGUAGE-SWITCHING. This is more common for websites that do not require big adaptations for different locales. Whenever the website for a certain culture is too different from other versions due to social, political, or commercial issues, this pattern may not be recommended. Web applications either adopt MULTILINGUAL USER PROFILE (Facebook, Twitter, etc.) or DYNAMIC CULTURAL PREFERENCES (Booking.com, Hostelworld, etc.) to allow users to switch the language at interaction time.

Mobile apps: A variation of UI LANGUAGE-SWITCHING has been adopted for the core of mobile operating systems development kits. Android and iOS applications provide different versions of localized resources, which are bundled together with the app installation package. Whenever the user switches the language of the mobile device in the MULTILINGUAL USER PROFILE, all apps switch their interface languages too. In case a certain app is not localized to the chosen language, the resource files will not be present for that specific locale and the interface will switch to a default language provided by the app. The default language of a multilingual mobile app is typically (but not necessarily) English. Besides that, the PREFERRED LANGUAGES HIERARCHY configured in a mobile device MULTILINGUAL USER PROFILE also have other consequences on the user experience: it can determine, for example, the different options of language/layout available for the virtual keyboard.

Desktop: Operating systems (OS) such as Windows and macOS allow users to choose a PREFERRED LANGUAGES HIERARCHY in their MULTILINGUAL USER PROFILE. Not all desktop applications for these systems, however, make use of that information. Some desktop applications will follow the OS configured locale, similar to what happens with mobile apps. Others present a behavior based on DYNAMIC CULTURAL PREFERENCES, resembling the behavior of some aforementioned websites.

3.1.6 *Consequences.*

Benefits:

- In websites, users can quickly switch the interface language without having to navigate to a new URL in the browser;
- In mobile apps, users can quickly switch the interface language by changing the list of preferred languages in the device configuration;
- In desktop applications, users can switch the interface language without having to download and install a new localized version of the application;
- Users can switch the language without noticing a drastic change in the interface layout and overall experience.

Liabilities:

- The website or application tends to become more uniform among all locales, which is not always desirable. Depending on the application domain, some cultures may require bigger adaptations in the user interface, which can be rendered more difficult or even impossible when using this pattern.

- The restrictions imposed by the mobile operating systems can limit the developers in the ways to implement this pattern.
- Developers who adopt this pattern must be careful about maintaining the state of the session when changing the language/locale of the interface, ensuring that users do not have to restart their current task all over again.

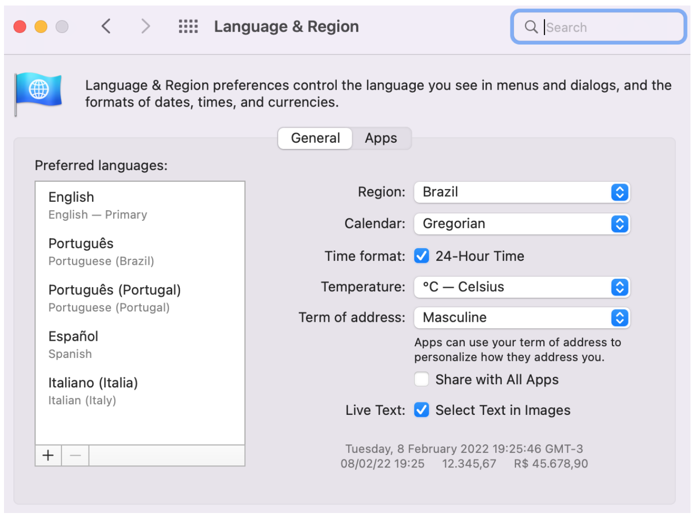
3.1.7 Known uses. The UI LANGUAGE-SWITCHING is part of the default architecture of most mobile operating systems, including Android and iOS. All mobile apps that include localized resources for two or more locales can be said to implement this pattern. This pattern is common in accommodation websites such as Booking.com and Hostelworld. It is also common in social networks such as Facebook and Twitter.

3.1.8 Related patterns.

- Applications that implement this pattern may rely on MULTILINGUAL USER PROFILE to allow users to choose their PREFERRED LANGUAGES HIERARCHY.
- Systems may also allow the selection of the interface language by means of DYNAMIC CULTURAL PREFERENCES configured through an INTERFACE LANGUAGE SELECTOR.

3.2 MULTILINGUAL USER PROFILE

Users are allowed to inform cultural preferences in their user profiles. Fig. 4 shows an example of the MULTILINGUAL USER PROFILE pattern as seen in macOS 12.0.1.



Language & Region preferences control the language you see in menus and dialogs, and the formats of dates, times, and currencies.

Preferred languages:

English
English — Primary
Português
Portuguese (Brazil)
Português (Portugal)
Portuguese (Portugal)
Español
Spanish
Italiano (Italia)
Italian (Italy)

Region: Brazil
Calendar: Gregorian
Time format: ✓ 24-Hour Time
Temperature: °C — Celsius
Term of address: Masculine
Apps can use your term of address to personalize how they address you.
☐ Share with All Apps
Live Text: ✓ Select Text in Images

Tuesday, 8 February 2022 19:25:46 GMT-3
08/02/22 19:25 12.345,67 R$ 45.678,90

Fig. 4. Example of the MULTILINGUAL USER PROFILE pattern in macOS 12.0.1.

3.2.1 Context. The developer wants to implement a user profile in a DESIGN FOR MULTILINGUALS. This is one of the strategies to allow users to choose the interface language in a UI LANGUAGE-SWITCHING.

3.2.2 Problem. When configuring their user profiles, multilingual users may feel the necessity to inform their cultural preferences, particularly preferred languages, besides regular personal information. These preferences may include competences in multiple languages, preferred timezone, preferred currency, among others.

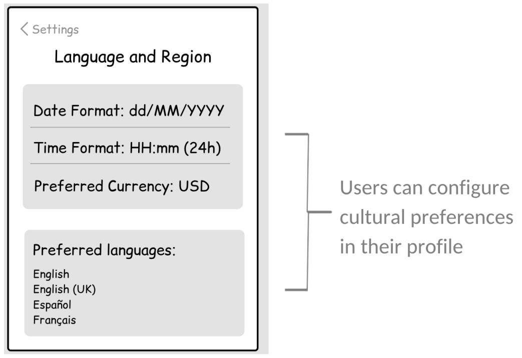
3.2.3 Forces.

- In a multicultural world, users may have specific cultural preferences, not always addressed by regular choices associated to a single culture/locale.
- Some users find it annoying to have to configure additional items in their profile.

3.2.4 Solution. **Allow users to inform cultural preferences in their user profiles.** The interface to configure the user profile should be extended with fields specific for cultural preferences, particularly the preferred languages. Some typical preferences that can be configured are:

- Competences in multiple languages;
- Preferences related to content translation;
- Timezone;
- Currency;
- Number, date, and time formats;
- Measurement units: Metric System, Imperial System, etc.

According to the domain of the system, the user profile can contain one or more of the aforementioned preferences. Allowing the configuration of multiple languages is the only one that is mandatory. The system should also provide reasonable default values so that the user does not have the necessity to fill all the items when changing the locale. Fig. 5 shows a proposed diagram for the MULTILINGUAL USER PROFILE pattern.



Date Format: dd/MM/YYYY
Time Format: HH:mm (24h)
Preferred Currency: USD

Preferred languages:
English
English (UK)
Español
Français

Fig. 5. A diagram for the MULTILINGUAL USER PROFILE pattern.

3.2.5 Consequences.

Benefits:

- By using this pattern, users can fine tune their cultural preferences, without being stuck with a single locale setting.
- Default values for specific locales speed up the configuration of the user's preferences.

Liabilities:

- More complexity is added to the system in order to cope with the additional preferences.
- The preferences configured using this pattern should be reflected in the rest of the system in order to fulfill user's expectations.

3.2.6 *Known uses.*
This pattern can be found in most current operating systems, including Windows, macOS, Android, and iOS. It can also be found in the configuration of Web platforms such as Google and Facebook.

3.2.7 *Related patterns.*

- A PREFERRED LANGUAGES HIERARCHY is the typical widget used to allow language selection in a MULTILINGUAL USER PROFILE.
- The information provided in this type of profile, especially the competences in multiple languages, can impact many other elements of the DESIGN FOR MULTILINGUALS, including DYNAMIC CULTURAL PREFERENCES, MULTILINGUAL OUTPUT, MULTILINGUAL INPUT, and CROWDSOURCED TRANSLATION.

3.3 PREFERRED LANGUAGES HIERARCHY

Widget used by multilingual users to inform to the system their preferred languages for data input/output. On the left part of Fig. 4 there is an example of the PREFERRED LANGUAGES HIERARCHY pattern as seen in the MULTILINGUAL USER PROFILE of macOS 12.0.1.

3.3.1 *Context.*
Used in the implementation of MULTILINGUAL USER PROFILE.

3.3.2 *Problem.*
Multilingual users need to inform a list of preferred languages in their profile, ideally expressing the order of preference between the languages.

3.3.3 *Forces.*

- Most users exhibit some level of multilingualism and want to inform their language competences instead of setting up a single locale;
- Among the user's informed languages, one is preferred over the others (typically, but not necessarily, their mother tongue).
- There is usually an order of preference among languages chosen by the user (typically, but not necessarily, reflecting their level of fluency in each language).
- Users expect the system's interface to adapt accordingly to a multilingual setup.

3.3.4 *Solution.*
Provide a compound widget that displays the preferred languages of the user and allows the modification of that list. The languages in the list are organized in a descendant order meaning that the most important languages will be displayed first. In most cases, the order of the languages will reflect how fluent the user is in each one of them (even though cultural and economic factors, among others, can also influence this decision). As a result, the first language is usually the mother tongue of the user and also the language used for the display of static content within the interface (the *interface language*). Highlighting the fact that the first language is the main interface language is recommended.

Languages should be differentiated by their dialects or locale variations. For example, making a clear distinction between *Portuguese* (international Portuguese), *Brazilian Portuguese*, and *European Portuguese* (Portuguese from Portugal). Language names should be displayed both in their original/native language and in the current interface language. Allow users to change the position of the languages and to remove languages from the list. Once the user correctly fills in his/her preferred languages, the system can make varied use of this information. This kind of configuration contrasts with single-locale configuration, in which users are asked for a locale (country and dialect) and all other cultural preferences are automatically determined including the interface language. Fig. 6 shows a proposed diagram for the PREFERRED LANGUAGES HIERARCHY pattern.

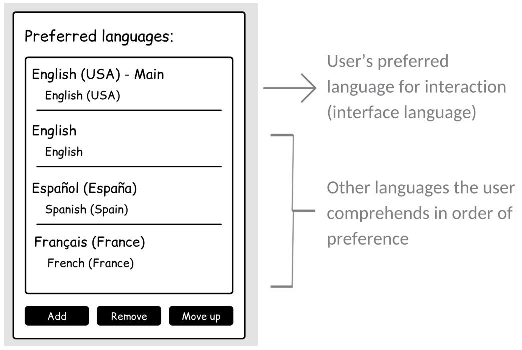


Fig. 6. A diagram for the PREFERRED LANGUAGES HIERARCHY pattern.

3.3.5 *Consequences.*

Benefits:

- Users are allowed to input a set of preferred languages, better representing their language competences.
- Among all list items, users are allowed to inform one favorite language that is to be used as interface language.
- The organization of the language hierarchy represents the user's order of preference.
- Developers have richer information about the users' language competences, which can be used throughout the system.

Liabilities:

- This pattern can put pressure on the developers, forcing them to adapt the system to consider all the new language configurations. Not taking this setup into account could disappoint the user's expectations.

3.3.6 Known uses. This pattern can be found in most current operating systems, including Windows, macOS, Android, and iOS. It can also be found in the configuration of Web platforms, such as Google Accounts, and Web browsers such as Google Chrome.

3.3.7 Related patterns. A widget similar to a INTERFACE LANGUAGE SELECTOR can be used when users need to add a new language to the hierarchy.

3.4 DYNAMIC CULTURAL PREFERENCES

Widgets in the application screens/pages make it possible to change cultural preferences at any time. Fig. 7 shows an example of the DYNAMIC CULTURAL PREFERENCES pattern as seen on the Hostelworld website.

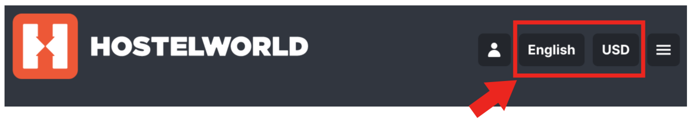


Fig. 7. Example of DYNAMIC CULTURAL PREFERENCES as seen on the pages of Hostelworld (September 2022).

3.4.1 Context. A DESIGN FOR MULTILINGUALS that allows quick changes of cultural preferences. This is one of the strategies to allow users to choose the interface language in a UI LANGUAGE-SWITCHING.

3.4.2 Problem. Multilingual users may need to quickly change the language and other cultural preferences at interaction time to facilitate the understanding and/or navigation of the displayed information.

3.4.3 Forces.

- Varied situations during the interaction can influence users to change the language or the currency used in the pages/screens of an application.
- Users may feel the necessity to change a cultural preference just for a while and then switch back to the previous setting.

3.4.4 Solution. **Offer, by means of widgets on the screens/pages of the application, the possibility to change cultural preferences at any time during the interaction according to the user's will.** The most frequent preferences addressed in implementations of this pattern are interface language and currency. Fig. 8 shows a proposed diagram for the DYNAMIC CULTURAL PREFERENCES pattern.

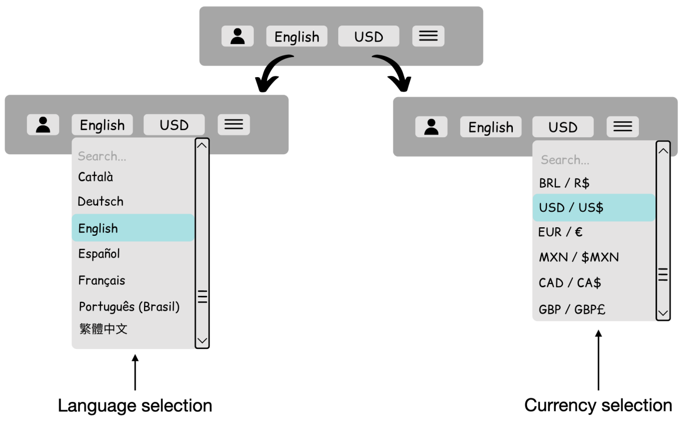


Fig. 8. A diagram for the DYNAMIC CULTURAL PREFERENCES pattern.

3.4.5 Consequences.

Benefits:

- Users can change language or currency of the application without having to access the user profile or system settings.
- Changing these settings and then switching back to the previous values is facilitated by widgets on all pages/screens of the application.

Liabilities:

- To be effective, this pattern must be present in all or almost all pages/screens of the application, adding complexity to the system.

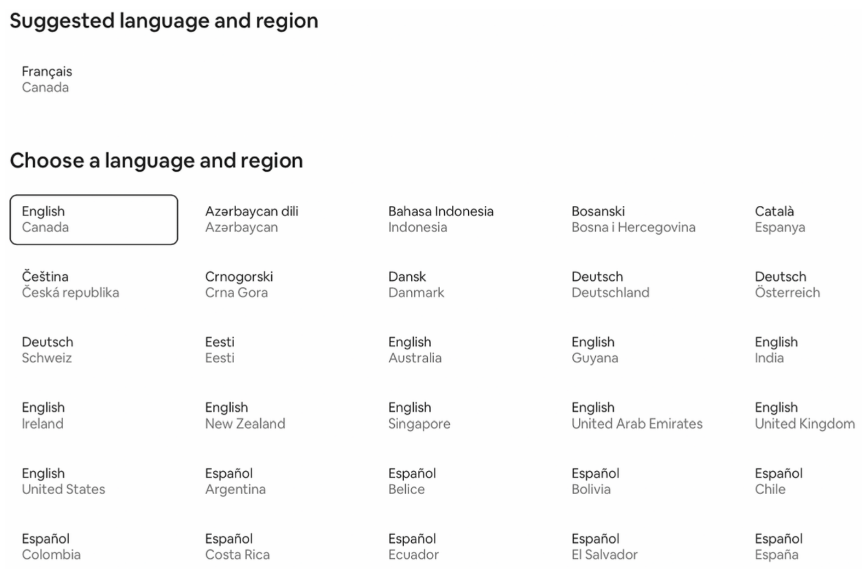
3.4.6 Known uses. This pattern can be found in a number of travel and tourism websites, including the ones from accommodation (Booking.com, Hostelworld, Hotels.com, etc.) and airline companies (Lufthansa, Ryanair, etc.).

3.4.7 Related patterns.

- The dynamic configuration of the interface language is typically implemented with an INTERFACE LANGUAGE SELECTOR.
- CURRENCY SELECTOR is the widget used to dynamically configure the currency of the application.

3.5 INTERFACE LANGUAGE SELECTOR

A widget that allows multilingual users to choose in which language the system is to be presented. Fig. 9 shows an example of the INTERFACE LANGUAGE SELECTOR pattern as seen on the Airbnb website.



Suggested language and region

Français
Canada

Choose a language and region

English Canada	Azərbaycan dili Azərbaycan	Bahasa Indonesia Indonesia	Bosanski Bosna i Hercegovina	Català Espanya
Čeština Česká republika	Crnogorski Crna Gora	Dansk Danmark	Deutsch Deutschland	Deutsch Österreich
Deutsch Schweiz	Eesti Eesti	English Australia	English Guyana	English India
English Ireland	English New Zealand	English Singapore	English United Arab Emirates	English United Kingdom
English United States	Español Argentina	Español Belice	Español Bolivia	Español Chile
Español Colombia	Español Costa Rica	Español Ecuador	Español El Salvador	Español España

Fig. 9. Example of the INTERFACE LANGUAGE SELECTOR pattern as seen in AirBnb (February 2022).

3.5.1 Context. Used in the implementation of DYNAMIC CULTURAL PREFERENCES in a DESIGN FOR MULTILINGUALS.

3.5.2 Problem. Users need a way to change the interface language at any time during the interaction.

3.5.3 Forces.

- Multilingual users may require to change the interface language frequently, while monolinguals would rather not bother about this issue.

3.5.4 Solution. **Provide a widget with which the user can easily change the interface language.** To define an initial or default value, the system can infer the locale from the IP address of the current connection or from previous user settings/activity. RECOMMENDED LANGUAGES can be used to provide suggestions for the most used languages and facilitate user interaction. As soon as the system receives the command, it must convert the entire interface to the selected language. Fig. 10 shows a proposed diagram for the INTERFACE LANGUAGE SELECTOR pattern.

In the presentation of the available options, country flags can be used to represent locales (FLAGS FOR COUNTRIES), but should be avoided to represent languages for two reasons: many countries have more than one official language; and many languages span more than one country. In some cases, the adoption of flags to represent languages can be offensive to members of certain communities. Regional language dialects should be differentiated by appending the name or flag of the country (for example, American English is typically presented as *English (USA)*).

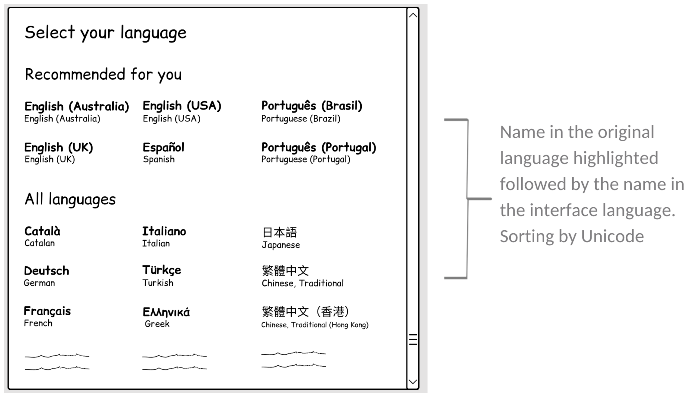


Fig. 10. A diagram for the INTERFACE LANGUAGE SELECTOR pattern.

3.5.5 *Implementation.*

Web applications: In websites, an overlay menu is typically used to present the list of available languages/locales. Presenting the languages in a grid like structure can be helpful to take advantage of the screen space and reduce the necessity of scrolling.

Mobile apps: Changing the language of mobile devices (through the operating system configuration) may also change the language of all running apps. In cases there is little screen space (e.g., in smartphones), the INTERFACE LANGUAGE SELECTOR can be presented in a compact form. Typically the language list is presented in a linear form, forcing the user to scroll the list to find the desired language. Recommended languages may be shown at the top of the list to diminish the probability of scrolling. To save space, the names of the languages are displayed only in their original (native) language and not on both native and interface language.

3.5.6 *Consequences.*

Benefits:

- This pattern allows multilinguals to change the interface language at any time and still addresses monolinguals by conveniently providing a default interface language.

Liabilities:

- In cases the list gets too big, users might find it difficult to pick the desired language.

3.5.7 Known uses. This pattern can be found in a number of accommodation websites such as Booking.com, Hostelworld, and Hotels.com.

3.5.8 *Related patterns.*

- In e-commerce applications, this pattern is frequently used in combination with a CURRENCY SELECTOR.
- RECOMMENDED LANGUAGES can be used to help users find the desired language faster.

- This pattern is similar to the GLOBAL GATEWAY as presented in the book by Junker [12], even though the problems addressed by each of these patterns are slightly different.

3.6 RECOMMENDED LANGUAGES

The system tries to recommend to the user the most probable language choices. An example of this pattern can be seen on the top section of Fig. 9 as part of the INTERFACE LANGUAGE SELECTOR found in the Airbnb website.

3.6.1 Context. This pattern usually appears in the implementation of INTERFACE LANGUAGE SELECTOR.

3.6.2 Problem. When facing a big list of languages/locales, users may find it difficult to find their desired option.

3.6.3 Forces.

- User are more prone to select certain languages over the others.

3.6.4 Solution. **Together with the complete list of languages/locales available for one application, display a selected list of recommended languages.** The list of recommendations does not necessarily follow any sort of order, and can be suggested using algorithms that infer the most probable languages. The recommendation algorithm may take into consideration available data, such as: preferences set up in the MULTILINGUAL USER PROFILE(), website usage statistics, history of user interactions, and so forth. Fig. 11 shows a proposed diagram for the RECOMMENDED LANGUAGES pattern.

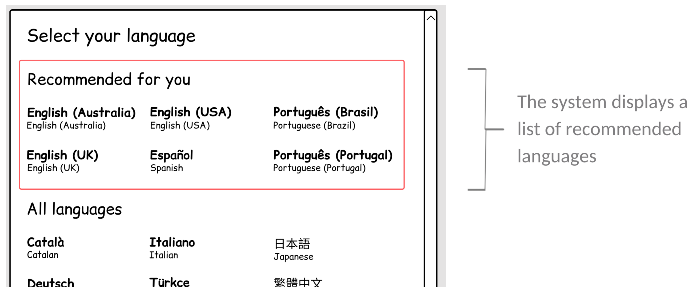


Fig. 11. A diagram for the RECOMMENDED LANGUAGES pattern.

3.6.5 Consequences.

Benefits:

- This pattern speeds up the process of language selection, by presenting a list of recommended languages.

Liabilities:

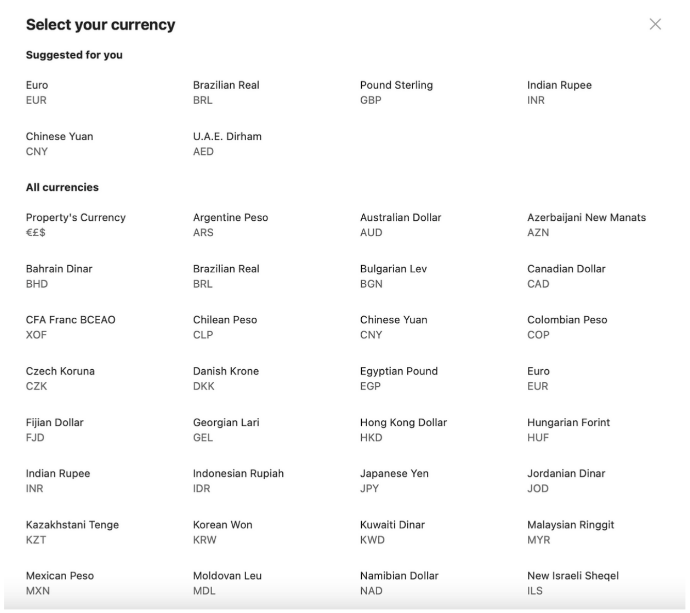
- A bad algorithm can recommend languages that are not the most appropriate for a certain user in a certain situation, hindering the user experience instead of facilitating it.

3.6.6 Known uses. This pattern can be found in a number of accommodation websites such as Airbnb, Booking.com, and Hostelworld.

3.6.7 Related patterns. This pattern is typically found in the implementation of INTERFACE LANGUAGE SELECTOR, but could also be used in other widgets that present lists of languages/locales, such as PREFERRED LANGUAGES HIERARCHY.

3.7 CURRENCY SELECTOR

A widget that allows multilingual users to choose in which currency the system should present monetary values. Fig. 12 shows an example of the CURRENCY SELECTOR pattern as seen on Booking.com.



Select your currency

Suggested for you

Euro EUR	Brazilian Real BRL	Pound Sterling GBP	Indian Rupee INR
Chinese Yuan CNY	U.A.E. Dirham AED		

All currencies

Property's Currency €£$	Argentine Peso ARS	Australian Dollar AUD	Azerbaijani New Manats AZN
Bahrain Dinar BHD	Brazilian Real BRL	Bulgarian Lev BGN	Canadian Dollar CAD
CFA Franc BCEAO XOF	Chilean Peso CLP	Chinese Yuan CNY	Colombian Peso COP
Czech Koruna CZK	Danish Krone DKK	Egyptian Pound EGP	Euro EUR
Fijian Dollar FJD	Georgian Lari GEL	Hong Kong Dollar HKD	Hungarian Forint HUF
Indian Rupee INR	Indonesian Rupiah IDR	Japanese Yen JPY	Jordanian Dinar JOD
Kazakhstani Tenge KZT	Korean Won KRW	Kuwaiti Dinar KWD	Malaysian Ringgit MYR
Mexican Peso MXN	Moldovan Leu MDL	Namibian Dollar NAD	New Israeli Sheqel ILS

Fig. 12. The CURRENCY SELECTOR as seen on Booking.com (March 2022).

3.7.1 *Context.*
This pattern is usually part of DYNAMIC CULTURAL PREFERENCES in e-commerce applications.

3.7.2 *Problem.*
In e-commerce systems where international purchases are frequent, users may feel the necessity to change the currency in which the prices of products/services are displayed.

3.7.3 *Forces.*

- Users that perform frequent purchases in international websites may require to change the currency frequently.

3.7.4 *Solution.*
Provide a widget with which the user can easily switch between currencies. To define an initial or default value, the system can infer the locale from the IP address of the current connection or from previous user settings/activity. Suggestions of the most frequently used currencies may speed up the selection process. When the system receives the currency change command, the platform must quickly convert the displayed prices and other monetary values to the selected currency. Just as the INTERFACE LANGUAGE SELECTOR, this pattern can also be presented in a compact form when displayed in smaller screens. Fig. 13 shows a proposed diagram for the CURRENCY SELECTOR pattern.

3.7.5 *Consequences.*

Benefits:

- This pattern allows international users to change the interface currency at any time and still addresses co-nationals that opt to keep up with the default currency.

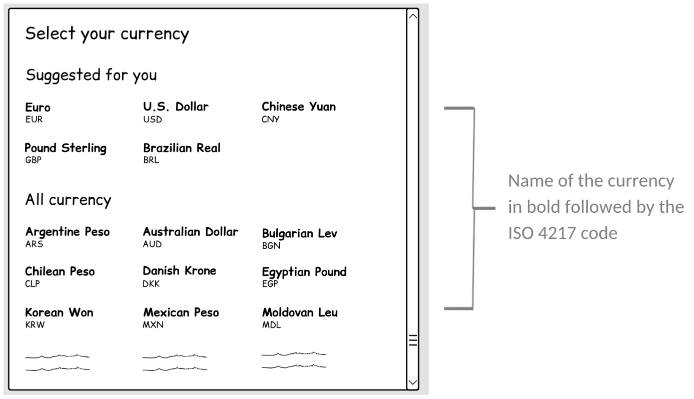


Fig. 13. A diagram for the CURRENCY SELECTOR pattern.

Liabilities:

- In cases the list gets too big, users might find it difficult to pick the desired currency.
- Deciding to adopt such pattern may impose difficult implementation decisions to developers, such as obtaining the most accurate exchange rates and performing the right exchange calculations.

3.7.6 *Known uses.*

This pattern can be found in accommodation websites such as Booking.com and Hostelworld.

3.7.7 *Related patterns.*

This pattern is often used in combination with a INTERFACE LANGUAGE SELECTOR.

4 CONCLUSION AND FUTURE WORK

The awareness of an increasing multilingual user base of software systems is motivating designers to adapt user interfaces to bilingual and multilingual users. In this paper we presented the evolution of a pattern language, providing descriptions for a new pattern and for six remodeled patterns. Future work include the validation of the language through empirical methods.

REFERENCES

[1] Jan O. Borchers. 2000. A Pattern Approach to Interaction Design. In *Proceedings of the 3rd Conference on Designing Interactive Systems (DIS).* ACM, New York, 369–378. https://doi.org/10.1145/347642.347795

[2] Diego Moreira da Rosa, Natanael Kuniechik, Soraia Raupp Musse, and Milene Selbach Silveira. 2021. Analyzing the Presentation of Multilingual User Reviews in Accommodation Websites. In *Proceedings of the XX Brazilian Symposium on Human Factors in Computing Systems (IHC).* 1–7. https://doi.org/10.1145/3472301.3484333

[3] Diego M. Da Rosa and Lucas P. Pons. 2017. Evaluating the presentation of user reviews in multiple languages: a semiotic approach. In *Proceedings of the 8th Latin American Conference on Human-Computer Interaction (CLIHC).* 1–8. https://doi.org/10.1145/3151470.3151472

[4] Diego Moreira Da Rosa, Maria Luisa Lamb Souto, and Milene Selbach Silveira. 2022. Designing interfaces for multilingual users: a pattern language. In *25th European Conference on Pattern Languages of Programs (EuroPLoP).* To be published.

[5] Clarisse Sieckenius De Souza. 2005. *The semiotic engineering of human-computer interaction.* MIT press.

[6] David M. Eberhard, Gary F. Simons, and Charles D. (eds.) Fennig. 2022. *Ethnologue: Languages of the World* (twenty-fifth ed.). SIL International. http://www.ethnologue.com

[7] Scott A. Hale. 2014. Design for multilinguals: Seemingly simple yet often missed. Oxford Internet Institute Blog. https://www.oii.ox.ac.uk/blog/design-for-multilinguals-seemingly-simple-yet-often-missed/ (accessed in June 2022).

[8] Robert B. Kaplan and Richard B. Baldauf. 1997. *Language planning from practice to theory.* Multilingual Matters, Bristol.

[9] Suzanne Romaine. 1995. *Bilingualism* (2nd ed.). Wiley-Blackwell.

[10] G. R. Tucker. 1998. A global perspective on multilingualism and multilingual education. In *Beyond bilingualism: multilingualism and multilingual education*, Jasone Cenoz and Fred Genesse (Eds.). Multilingual Matters, Bristol, Chapter 1, 3–15.

[11] Li Wei. 2007. Dimensions of bilingualism. In *The bilingualism reader*, Li Wei (Ed.). Routledge, Abingdon/New York, Chapter 1, 3–22.

[12] John Yunker. 2010. *The Art of the Global Gateway: Strategies for successful multilingual navigation.* Byte Level Books.