Global local

Making a localisation interface work on a grand scale so that it can be applicable across different languages

Localisation is a growing area for many translators. However, that's often on a relatively small scale. In 2018, the language services department at our firm Avaloq (which works in digital banking solutions, core banking software and wealth management technology) was approached by the in-house team developing the firm's online banking application. Our colleagues had developed the interface in English and needed it to be localised for the French-, Italian- and German-speaking markets. They had already translated part of the user interface into these languages, but needed a review of the English and of the existing translations as well as delivery of the remaining translations.

Beginning at the beginning

Originally, at the start of each project, the developers exported the complete collection of translatable text (strings) from their development stack and delivered it to us in a large multilingual Excel file containing all the languages, with English in the first row, then German, French and Italian underneath. This was repeated for every single user interface (UI) string. Sometimes, the files contained as many as 24,000 rows, which turned coordinating the work in the different languages into an absolute nightmare.

The complete lack of context also proved problematic for the reviewers and translators. How do you translate 'Open' to German if you have no idea where it appears on the web banking user's screen and whether it is an adjective or a verb? *Offen*? *Öffnen*? At some point, a trial online banking environment was made available to give us some context, although there was no way to verify if you had located the actual text in question, because the keys were not displayed anywhere in that environment. The large multilingual Excel file with its basic search and comment functions also made it difficult and time-consuming to ensure consistency and maintain an overview of the whole project. Using our CAT tool for translating was not an option either, because the strings in the four languages were not listed in separate columns, but underneath each other in blocks of four rows for every single string. Converting the Excel file to a format that memoQ could handle would have been highly complex, if not impossible.

After working on a few projects with these limited tools, we expressed our concerns to the online banking team. Fortunately, the team members understood the challenges we faced and in fact they were redesigning their product and development processes. They assessed the features of the potential solutions, seeing which would suit our needs best.

How do you translate 'Open' to German if you have no idea where it appears on the web banking user's screen and whether it is an adjective or a verb?

A new and more workable translation centre

The solution they chose is a cloud-based platform that offers translators and translation project managers a whole host of practical features to enhance their localisation work, from an in-context editor to a comment function and translation memories and glossaries. When we started using it, we immediately noticed the improvement in our localisation process. The platform's translation centre is the main area where translators and reviewers work and has a really user-friendly layout and design. The platform also enables two different workflows: the basic workflow covers the statuses 'untranslated', 'translated' and 'unverified' (where the source string has been changed after it was translated), and the advanced workflow incorporates a review process too. These statuses are colour-coded and appear very clearly beside all strings listed to the left of the editable target text field.

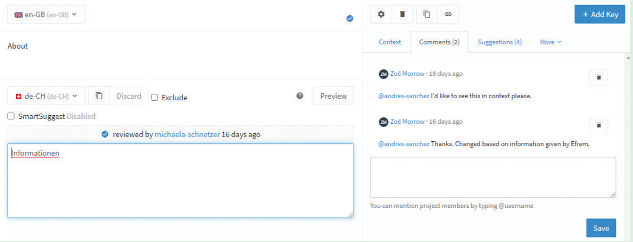
The list includes a filter function that allows us to display strings based on their status, so we can focus on only 'unreviewed' ones, for instance, and filter out strings that have already been reviewed and finalised.

The lack of context we previously struggled with has mostly become a thing of the past, as the in-context editor allows us to actually locate and see the strings in the context of the online banking user interface. We can choose to 'test-drive' our online banking application and translate and review text directly there, before returning to the translation centre to work on text that didn't crop up as we navigated through the application. (In theory, we can also look at the online banking application and see the strings in situ, but in reality we haven't managed it yet.)

Another helpful feature is the translation memory, which enables some time savings. As a customer account can contain an unlimited number of localisation projects, you can choose which projects should use the account-wide translation memory and which should be limited to their local (project-wide) one. The previous translations in the memory appear under the 'Suggestions' tab on the right of the translation centre, so we can see exactly how and when the source text was translated for all previous occurrences in our project or even a different project.

We can also insert the previous translation into the target field by simply clicking on ‘Apply’. As the platform allows you to import TMX files from your usual CAT tool, we made the effort to align our previous localisation projects. So, thankfully, gone are the days of ‘Ctrl+F’ searching through the Excel sheets of previous localisation orders for online banking!

ICU message format



☐ SmartSuggest Disabled

reviewed by michaela-schnetzer 16 days ago

Informationen

Zoë Morrow · 18 days ago

@andres-sanchez I'd like to see this in context please.

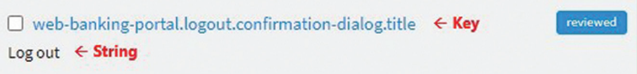
Zoë Morrow · 16 days ago

@andres-sanchez Thanks. Changed based on information given by Efrem.

You can mention project members by typing @username

A format that allows developers to use dynamic values in their strings in the form of placeholders, such as for a person’s name, plural forms, currencies, etc. It is particularly useful for ensuring correct agreement in singular and plural forms in English, and also for cases and genders in other languages.
Raw: {count, plural, one{{count} client selected} other{{count} clients selected}}
Singular: {count} client selected Plural: {count} clients selected
Displayed in UI: 1 client selected 3 clients selected

Keys and strings



☐ web-banking-portal.logout.confirmation-dialog.title ← Key

Log out ← String

These are the building blocks of software localisation projects. Keys are unique IDs for each piece of text that appears on a user interface. The piece of text is referred to as a ‘string’.

Plug-ins and other practical considerations

We have also benefited from the time savings of using the MT plug-in options available for the platform. Having previously decided to roll out DeepL Pro as a machine translation solution to our colleagues across the company, we naturally opted for the DeepL plug-in. (See more on this in our feature in the March-April *Bulletin*.) This means that our translators can see the source of a translation at all times, as it is clearly marked above the target text field in the translation centre as having been machine-translated, auto-propagated from the translation memory, or indeed translated or reviewed by a user.

Other practical features that have improved our localisation process include a screenshot function in the ‘Context’ tab, where users can upload a screenshot of the string in context – this is particularly useful for localising mobile apps because the in-context editor is not available for this technology. There is a ‘History’ tab where we can check all activity on a key, including checking who did that activity, from when it was first imported into the platform.

We can also tag keys, which serves many different functions (such as version management for developers), and allows us to mark keys as having unclear text, call them up in a list by searching for the tag and then, for instance, address them one by one at a meeting with the developers. The preview function for messages in ICU format lets us check that the different placeholders work in the various languages, and the validation function clearly warns us in red text if we broke the message syntax as we were translating or reviewing the string.

Easier to work with colleagues

The greatest benefit we’ve noticed has been in the improved collaboration between our language services team and the developers. Even with the additional context the in-context editor supplies, we sometimes need to clarify strings that the developers have written; and previously we’d send over all our queries together, even though we needed answers from a host of different developers. Unsurprisingly, it was hard to coordinate, and some questions never got an answer. Now, we can simply use the comment function to direct questions to the developers by tagging them. They automatically receive an email notification with a link to our comment in the translation centre and can reply directly to us to explain. This has hugely improved communication and collaboration between the developers and our team and helps us to guarantee clarity and consistency in the source text: which means a more accurate translation in the target language.

> The greatest benefit we’ve noticed has been in the improved collaboration between our language services team and the developers

Of course, it hasn’t all been plain sailing. We still have issues with the in-context editor and establishing a link for every key between it and the translation centre. There also doesn’t appear to be a way to prevent the machine translation service from translating all parts of a string in ICU format (for instance, content in curly brackets), which means that machine-translated keys using ICU syntax always need to be fixed by the translators.

All in all, our shift to a localisation platform has proved to be a very positive experience. Having piloted it for our online banking team almost a year ago, their developers are pleased that we can now translate new features within a shorter time span and to a higher quality before each new version and collaborate with them in real time. We can even fix customer bugs directly in the platform under the developers’ guidance.

It’s exciting for us to see how the use of this platform has since expanded to other Avaloq applications. But the most satisfying part of this localisation journey, in addition to the great user experience, is witnessing how technological advancement is continuing to improve language services overall. It is a very inspiring time for our sector.

The Language Services team at Avaloq is Zoë Morrow, Deputy Head of Language Services, works from German, French and Spanish to English; Michaela Schnetzer, Language Services Specialist, works from English and French to German; Carina Freitag, Head of Language Services, works from English and French to German; Emma Gledhill FITI, Language Services Specialist, works from German, French and Dutch to English; and Esther Scowen-Bolz, Language Services Specialist, works from English and French to German. The team members have nearly a century of experience between them, which they draw on to provide translations and to proofread company documents. They also curate the corporate terminology, provide language consultancy for natural language processing (NLP) and localisation projects, and have implemented and use a wide range of language technology solutions.