

## Localisation

### De meilleurs résultats pour moins d'efforts grâce au partage des ressources

En 2001, une équipe de plus de 30 programmeurs avait pour mandat de développer une grosse application (le développement continue de nos jours) destinée à remplacer plus de 20 systèmes existants.

La livraison de la première version devait absolument respecter la date prévue, et le temps filait vite.

Le Bureau de la traduction ne peut vraiment pas se permettre de livrer dans une langue seulement. La Loi sur les langues officielles exige que les documents et les applications soient livrés simultanément en anglais et en français.

À cette époque, on trouvait des gens qui étaient fiers de dire que leur processus de localisation avait permis de n'attendre qu'un mois après la fin du développement pour la traduction. Nous avons plutôt décidé que la localisation se ferait **pendant le développement**, et de suivre la terminologie et les définitions à partir du début. Les programmeurs pouvaient consulter les langagiers pour mieux maintenir la terminologie, les traducteurs pouvaient consulter les programmeurs pour mieux traduire, ce qui rendait le processus plus fluide.

À partir du moment où les programmeurs ont cessé de travailler à la phase 1, il a fallu 12 heures pour livrer simultanément dans les deux langues officielles. On doit ce succès à une collaboration entre les programmeurs et les traducteurs axée sur l'utilisation commune d'un outil de gestion de la terminologie.

Les programmeurs inscrivaient le nom de toute composante textuelle de l'application (boutons, titres, menus, etc.) dans l'application de gestion de terminologie. Ils indiquaient ensuite s'il fallait traduire, réviser ou réviser et traduire. Quelques lignes de programme permettaient de générer l'interface à volonté.

Auparavant, il avait été convenu de respecter les principes d'internationalisation qui s'appliquaient ici (laisser de l'espace pour le français qui est plus long que l'anglais, ne jamais fusionner des messages, privilégier les messages spécifiques aux messages génériques, etc.). Les programmeurs pouvaient aussi inscrire l'information utile destinée aux traducteurs qui fait si souvent défaut dans les tâches de localisation.

Les programmeurs pouvaient faire une petite recherche sur les éléments existants plutôt que de réinventer la roue. Les diverses méthodes de recherche permettaient de trouver rapidement. En procédant de la sorte, on évitait que les traducteurs n'aient à demander aux programmeurs quelle est la nuance subtile entre deux libellés quasi-identiques.

Le numéro de téléphone des auteurs était automatiquement consigné, on pouvait donc les rejoindre quand des clarifications étaient nécessaires. Les traducteurs n'eurent vraiment pas à le faire souvent.

Les programmeurs étaient parfois bilingues, ils pouvaient alors inscrire le texte dans les deux langues et préciser s'il y avait lieu de traduire, de réviser ou de réviser et traduire. Certains programmeurs savent qu'il vaut mieux réviser le texte même dans leur langue maternelle.

Les traducteurs pouvaient rapidement trouver le nouveau matériel à traduire, modifier les fiches et préciser s'ils avaient révisé, traduit ou traduit et révisé.

Un rédacteur faisait alors la rédaction à plein temps de la documentation. Il pouvait aussi utiliser la base de données terminologiques pour s'assurer que sa terminologie était uniforme, et inscrire les termes qui n'étaient pas déjà là.

On a aussi fait la traduction de la documentation en continu à l'aide d'une mémoire de traduction. L'effort total de traduction était un peu plus grand, mais on n'a pratiquement jamais eu à attendre en raison de la traduction.

On peut certainement dire que les coûts additionnels de traduction étaient inférieurs aux économies de temps mort du côté des programmeurs.

En fait, la traduction en continu de l'interface a même permis un certain contrôle de la qualité bilingue pendant le développement, puisque les traducteurs pouvaient voir l'application dans les deux langues en environnement de test.

Pendant mon séjour ici, j'essaierai certainement de voir dans quelle mesure les wikis pourraient nous aider à l'avenir à traiter de la sorte la documentation et la terminologie multilingue.

Quelques saisies d'écran suivent.



Project Authc

▼

- PDF reports
- Supplementary Specification
- UC - Administration
- UC - Availability table
- UC - Billing
- UC - Business tables
- UC - Calendar template
- UC - Client Account
- UC - Contact
- UC - Direct billing
- UC - Email
- UC - External Interface - Client Account
- UC - External Interface - Client agreements
- UC - External Interface - MGTC
- UC - Impromptu Report
- UC - Internal Organization
- UC - Interpretation
- UC - News
- UC - Order
- UC - Task
- UC - Timesheet
- UC - User registration
- UC - User Regist - Sc REgister online
- UC - Virus Scan

Instead of an Abbreviation, we have the ID of the string

**Abbreviations**

GUI\_000065

Context +

This list element is displayed on form data exceptions for the alternate province field.

TR - To be done	TR - Work done	Extract for	Date created	Date modified	<input checked="" type="checkbox"/> Verified
Revise and Translate	Translated	Resource Bundle	2001-11-08	2001-11-09	

Do not modify

Revise and Translate

Revise only

Translate only

Revised and Translated

Revised

Translated

Applets - Up/Down

CodeTable - Billing Activities

CodeTable - Classification

CodeTable - Client Account Types

CodeTable - Communication Modes

CodeTable - Contact Role Types

CodeTable - Countries

CodeTable - Degree of specialization

CodeTable - End Uses

CodeTable - Interpretation Order

CodeTable - Language Categories

CodeTable - Order Statuses

CodeTable - Province / State

CodeTable - Resource Types

CodeTable - Routing Language

CodeTable - Service Categories

CodeTable - Service items

CodeTable - Task statuses

CodeTable - Tmsmss Method

CodeTable - Unit of measure (UOM)

CodeTable - Unit Type

Do not extract