



Les modèles technologiques de la localisation

Cécile Martin
Université Rennes 2

 **Définition**


- Localisation : LISA (Localization Industry Standards Association) la définit comme l'adaptation linguistique et culturelle d'un produit à ses régions de distribution et d'utilisation.

Avant d'entrer en détails dans les modèles technologiques de la localisation, nous allons définir la localisation. LISA (Localization Industry Standards Association) la définit comme étant « l'adaptation linguistique et culturelle d'un produit à ses régions de distribution et d'utilisation ».

 **Que regroupe la localisation ?**

- Logiciel : interface, aide en ligne, documentation
- Web : contenu textuel, contenu multimédia et technologies embarquées

Il est possible de distinguer deux types majeurs de localisation : la localisation logicielle et la localisation Web. La localisation logicielle regroupe la fameuse 'triplette' : l'interface, l'aide en ligne, la documentation. La localisation Web regroupe le contenu textuel, le contenu multimédia (images, fichiers vidéo, fichiers audio) et les technologies embarquées.


 **Modèles - Historique**

- Débuts de l'informatique ⇔ pas de localisation
- Une seule version de produit distribuée et vendue à toutes les cibles (anglais)

Dans cette présentation, nous allons nous limiter aux modèles technologiques de la localisation des interfaces logicielle et Web puisque la localisation des aides en ligne ou de la documentation relève des mêmes méthodes et compétences que la traduction 'classique'.

Nous allons présenter un historique des modèles technologiques de la localisation, en précisant que tous ces modèles sont encore utilisés aujourd'hui à différents niveaux dans l'industrie.

Aux débuts de l'informatique, la localisation n'existait pas. Le public était très restreint, se composait principalement d'ingénieurs, et était anglophone (ou du moins le supposait-on). Les logiciels étaient développés en anglais (la plupart étaient développés et commercialisés par des éditeurs de logiciels nord-américains) et étaient vendus tels quels dans le monde entier.



Modèles - Écrasement

- Modèle par écrasement
- Duplication intégrale du code : adaptation du code et traduction du contenu
- Inconvénients :
 - ⇒ Délai de livraison entre les deux versions
 - ⇒ Maintenance difficile à gérer

Avec la démocratisation de l'outil informatique, le public devient rapidement demandeur d'interfaces qui lui sont adaptées linguistiquement.

La première réaction de 'localisation' consiste à écraser l'application source. Le code est intégralement dupliqué : le contenu textuel (chaînes à afficher) est traduit et le code est adapté (par exemple si l'outil sert à classer des éléments dans l'ordre alphabétique ou effectue des calculs monétaires).

Le résultat de cette méthode de localisation est un produit packagé pour chaque langue.

Les principaux inconvénients sont le délai de livraison (l'application source doit être véritablement finie avant de pouvoir commencer la traduction) et la difficulté de maintenance. En effet, ce sont des équipes différentes qui travaillent pour chaque cible, et tout le code est en dur, c'est à dire que l'on arrive à des produits réellement distincts en fonction du marché linguistique – sans même parler des releases correctives qui deviennent alors différentes en fonction de la cible.


L'avantage principal de cette méthode est que le produit cible est développé par une équipe de spécialistes locaux (linguistes + ingénieurs) ce qui permet d'espérer une qualité optimale.

Une autre méthode d'adaptation du produit apparaît presque en même temps que l'écrasement, mais est beaucoup moins utilisée : la méthode par ajout (Add-On). « Un seul produit, plusieurs langues ».

Avec un système de 'si...alors' on procède à une exécution de sections différentes de codes en fonction de la locale du système sur lequel s'exécute l'application.

Le principal inconvénient de la méthode par ajout est que le produit ne peut pas être compilé tant que la traduction n'a pas été faite, ce qui nous ramène au problème du délai de livraison. De plus, la taille de l'application peut vite devenir prohibitive.

Ces deux modèles que nous venons de détailler ont en commun, et c'est bien leur défaut, de



Modèles - Ajout

- Modèle par ajout
- Un seul produit : version multilingue du code
- Inconvénients :
 - ⇒ Délai de livraison du produit
 - ⇒ Taille de l'application

RENNES UNIVERSITÉ
2

Modèles – I18N

- Modèle "i18n"
- Internationalisation + localisation : appel dynamique à des ressources externes
- Internationalisation = code unique dépourvu de tout élément culturel ou linguistique (+ nombreuses autres préparations)
- Fichiers ressources par langue

reposer sur des applications ‘codées en dur’. Est donc apparu un nouveau modèle qui ne repose pas que sur une simple adaptation d’un produit, mais qui demande une réflexion en amont : une conception fondamentalement différente du produit.

L’internationalisation

L’internationalisation consiste à préparer le produit de façon à ce que le code ne contienne aucune spécificité linguistique ou culturelle : toute spécificité culturelle (couleurs, icônes, etc.) et toute chaîne de texte sont extraites du code vers des fichiers ressources externes.

Sans oublier que le produit doit également subir d’autres préparations : la taille des boîtes de dialogue doit être modifiable, le support de jeux de caractères doit être prévu, une allocation dynamique des tailles de mémoire pour les messages doit être réalisée, etc.

Le résultat de ce modèle combinant internationalisation et localisation est un code unique, dépourvu de tout élément culturel ou linguistique, associé à au moins autant de fichiers ressources¹ que de langues ciblées.

RENNES UNIVERSITÉ
2

Modèles – I18N

- Inconvénients :
 - ⇒ Coût de l’internationalisation
 - ⇒ Traduction difficile
 - ⇒ Wrong file / Wrong address = {wrong}+{resource}
 - ⇒ Êtes-vous sûr de vouloir supprimer ces 3 fichiers Excel ? = Êtes-vous sûr de vouloir supprimer ces {0} fichiers {1} ?

Cette internationalisation a cependant un coût de développement important. De plus, la traduction des éléments de l’application est rendue difficile par le manque de contexte dans ces fichiers ressources et par l’utilisation importante de la concaténation des chaînes et des variables. Par exemple, des messages comme "Wrong file" et "Wrong address" peuvent être générés par le même code du type {wrong}+{resource}. De même, un message « Êtes-vous sûr de vouloir supprimer ces 3 fichiers Excel ? » peut être généré par un code du type « Êtes-vous sûr de vouloir supprimer ces {0} fichiers {1} ? ». Au cours du développement, le nommage des variables doit donc être extrêmement bien fait et très strict afin de minimiser les difficultés de localisation par la suite.

¹ Les fichiers ressources peuvent être des fichiers en texte brut ou des fichiers binaires (compilés). Dans le cas de fichiers binaires, le localiseur dispose de deux solutions de travail : soit de récupérer le source et de recompiler après adaptation, soit d’utiliser des outils de localisation comme Catalyst.

RENNES 2
UNIVERSITÉ
NANTES
METROPOLE

Le cas particulier du Web

- Public beaucoup plus diversifié
- But plus informatif que le logiciel
- Nombreuses technologies embarquées et multimédia

Nous allons maintenant aborder le cas particulier du Web. On retrouve plus ou moins les mêmes méthodes dans les choix d'architecture d'un site Web multilingue, même si le travail est différent en fonction de la finalité du site.

Sur le Web, on a rapidement eu affaire à un public plus diversifié que les utilisateurs de logiciels.

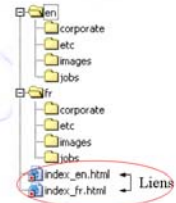
Dans le cadre de la localisation d'un site Web, il faudra distinguer le contenu à traduire du contenu nécessitant une adaptation régionale (offres spéciales, contacts commerciaux, etc.). Cette différenciation de type de contenu est plus rare dans les applications.

Les technologies embarquées (JavaScript, Java, Flash, etc.) se rapprochent de plus en plus de réelles applications et nécessitent également une réflexion d'internationalisation en amont.

RENNES 2
UNIVERSITÉ
NANTES
METROPOLE

Le cas particulier du Web

- Architecture "en miroir"
- Solution simpliste : souvent nécessaire si localisation non prévue en amont




Premier exemple d'architecture Web : une architecture en miroir qui consiste à dupliquer l'intégralité du site (pages et ressources multimédia). Cette architecture est souvent nécessaire lorsque la localisation n'a pas été prévue en amont.

RENNES 2
UNIVERSITÉ
NANTES
METROPOLE

Le cas particulier du Web

- Architecture "factorisée"
- Optimisation du volume du site multilingue

Choix de la langue



Un autre exemple d'architecture est une architecture factorisée qui consiste à ne dupliquer que les éléments spécifiques à la langue ou à la culture et à regrouper tous les éléments communs. Cette méthode permet d'optimiser le volume du site multilingue.

RENNES 2
UNIVERSITÉ
NANTES

Le cas particulier du Web

- Architecture "dynamique"
- Séparation de la forme et du contenu
- Génération dynamique des pages HTML à partir d'une base de données multilingue

Enfin, on trouve une architecture dynamique dans laquelle les pages présentées aux utilisateur sont générées dynamiquement à partir d'une base de données et de feuilles de style (Les formats CSS et MDB du schéma ne sont donnés qu'à titre d'exemple : cette génération dynamique est possible avec d'autres technologies, comme par exemple avec XML et XSL qui sont de plus en plus utilisés).

Cette architecture est aujourd'hui une véritable référence aujourd'hui en termes de localisation. Elle présente l'avantage de pouvoir rationaliser et simplifier les tâches de la localisation. Au traducteur, on fournit la base de données² qui contient tout le texte s'affichant dans les pages dynamiques, et aux graphistes on fournit les ressources multimédia. Par contre, cette architecture présente les même inconvénients que le modèle I18N logiciel : le manque de contexte de la base de données et l'utilisation éventuelle de la concaténation des chaînes et des variables.

Le localiseur doit être capable d'utiliser différents outils pour ces différentes méthodes :

RENNES 2
UNIVERSITÉ
NANTES

Outils

- Éditeurs de texte
- Studios de développement (IDE)
- Outils de localisation
- Outils de TAO
 - Importance d'utiliser une seule mémoire de traduction pour le projet global
 - Difficulté d'utilisation des outils de TAO classiques

- Des éditeurs de texte ou des studios de développement tels que Forte, Eclipse, Delphi, VisualBasic, .Net pour éditer le code source de l'application ;
- Des outils de localisation spécialisés pour travailler sur la localisation du code et/ou des fichiers ressources ;
- Des outils de TAO, en précisant l'importance d'utiliser une seule mémoire de traduction pour le projet global. De plus, il faut également noter que les outils de TAO classiques font perdre encore plus de contexte dans le cas de la localisation et peuvent rendre la traduction encore plus difficile.

² Base de données qui présente en plus l'énorme avantage de s'interfacer sans aucun problème avec tous les outils de TAO du marché.



Conclusion

- Toujours pas de recette magique conciliant coût minimal et qualité optimale
- Aujourd'hui, projets = cycles
- Nouvelles technologies et nouveaux standards (TV sur ADSL, applications mobiles, etc.)
- Envisager la localisation comme un nouveau métier de communication technique ?
- Enjeu technologique ou enjeu de formation ?

En conclusion, il faut malheureusement reconnaître que personne n'a trouvé la recette magique permettant de concilier un coût minimal à une qualité optimale.

Aujourd'hui, dans le cas d'applications logicielles ou Web, il n'existe plus réellement de date de début et de fin d'un projet de localisation, car les produits sont en constante évolution : le cycle de vie des applications logicielles a été considérablement réduit et se base souvent sur les releases correctives, alors que la qualité d'un site Web repose bien souvent sur la régularité de ses mises à jour. La localisation doit aujourd'hui être intégrée à ce cycle de vie toujours plus rapide.

Les nouvelles technologies qui apparaissent chaque mois (applications interactives de télévision sur ADSL, applications mobiles, etc.) représentent autant de nouveaux enjeux de localisation qu'il va falloir être capable d'adresser.

L'idéal serait peut-être que le localiseur partage le bureau des développeurs ainsi que certaines de leurs compétences. Ce système³ permet de réduire le délai de localisation au minimum et d'obtenir une qualité optimale.

Enfin, tous ces enjeux technologiques présents et futurs de la localisation représentent autant d'enjeux de formation à la traduction.

³ Ce type de poste est viable même pour des petites structures si le localiseur cumule des compétences de rédacteur technique, webmaster et/ou chef de projet (gestion de l'externalisation des autres langues, etc.).

