



**ASp**  
la revue du GERAS

11-14 | 1996  
Actes du 17<sup>e</sup> colloque du GERAS

---

## La fonction traduction en localisation logicielle

Uzoma Chukwu

---



### Édition électronique

URL : <http://journals.openedition.org/asp/3461>

DOI : 10.4000/asp.3461

ISSN : 2108-6354

### Éditeur

Groupe d'étude et de recherche en anglais de spécialité

### Édition imprimée

Date de publication : 1 décembre 1996

Pagination : 135-146

ISSN : 1246-8185

### Référence électronique

Uzoma Chukwu, « La fonction traduction en localisation logicielle », *ASp* [En ligne], 11-14 | 1996, mis en ligne le 29 avril 2013, consulté le 20 avril 2019. URL : <http://journals.openedition.org/asp/3461> ; DOI : 10.4000/asp.3461

---

Ce document a été généré automatiquement le 20 avril 2019.

Tous droits réservés

---

# *La fonction traduction en localisation logicielle*

Uzoma Chukwu

---

## Introduction

- 1 En arrêtant le titre de ma communication<sup>1</sup> sur la formulation ci-dessus, j'ai voulu poser d'entrée la problématique de ma contribution — qu'est-ce qu'est la localisation logicielle ? —, mais aussi préjuger de ce que sera ma position sur la question. Le terme est-il simplement une appellation glorifiée pour une activité qui se fait de tout temps, à savoir la traduction ? Quel est le rôle exact du traducteur en localisation ? Pour tenter de répondre à ces interrogations, je procéderai en présentant l'ensemble des tâches qu'on peut normalement envisager dans un projet de localisation pour mieux les répartir entre la fonction traduction et toute autre fonction partie prenante. Je serai également amenée à introduire deux autres notions, celle d'internationalisation et celle de globalisation, qui sont aujourd'hui incontournables dans les discussions sur la localisation, mais qui peuvent être source de confusion.

## 1. Qu'est-ce qu'est la localisation logicielle ?

- 2 Plutôt que d'aller chercher des définitions qui pourraient ne pas nous être d'un très grand secours à ce stade, je poserai le problème de la façon la plus concrète, en proposant deux cas. Dans la mesure où ces deux cas peuvent être considérés comme représentant deux extrêmes, ils doivent permettre d'évoquer la majeure partie des questions qui ont une place dans la discussion.
- 3 CAS n° 1 : La Société XYZ est l'un des plus grands noms en matière d'édition de logiciels. Il y a quelques mois elle travaillait encore sur un projet pour un système de gestion de base de données qui devait révolutionner ce qu'on sait faire en la matière, projet connu sous le nom de code Mars. Le savoir-faire, le renom et même l'hégémonie de XYZ sur la plupart

des fronts sont bien établis sur le plan international. Ainsi, le marché visé par le projet Mars dès le départ était le marché international, ce qui a influencé la conception et l'élaboration du logiciel, du début jusqu'à la fin. Juste un mois après le lancement officiel de la version internationale du logiciel (qui se trouve être la version américaine mais qui aurait pu aussi être la version française ou allemande, etc.<sup>2</sup>), les versions allemande, britannique, française, japonaise et, même, arabe étaient déjà annoncées.

- 4 CAS n° 2 : LogiMicro est une jeune société grenobloise d'édition logicielle créée par trois anciens camarades de classe qui en avaient assez de chercher du travail après leurs études. Conscients des difficultés aujourd'hui de s'introduire dans l'édition de logiciels où les lois sont dictées par quelques grands, les trois associés avaient décidé de se situer dans l'édition de logiciels spécialisés. Voici déjà trois ans qu'est sortie la première version de leur logiciel de commerce électronique qui a d'ailleurs remporté un tel succès que LogiMicro est aujourd'hui considérée par les commentateurs comme un cas d'école du dynamisme de l'industrie de l'édition de logiciels. Ils finalisaient les derniers détails pour la version 2.0 du logiciel lorsque la tentation de la conquête d'autres marchés, en commençant par le marché « anglo-saxon », les a pris. Ils ont pris contact avec un traducteur, lui ont expliqué leurs intentions et lui ont demandé de se mettre au travail. Il faudrait des pages entières pour relater ce qui s'est passé par la suite. Mais, pour aller vite, la version « anglo-saxonne » qu'ils espéraient sortir dans les deux mois n'a jamais vu le jour pour la simple raison qu'en la matière, il n'y a pas de marché anglo-saxon, donc impossible de concevoir une version anglo-saxonne. En revanche, après presque un an d'effort, ils ont fini par sortir nombreuses versions — américaine, britannique, canadienne, etc. — qui, tout en ayant une interface et une documentation en anglais, n'étaient en rien interchangeables. Nos trois associés ne sont pas pour le moment pressés de s'attaquer aux marchés sud-est asiatiques, moyen orientaux, etc. parce qu'ils savent qu'il leur faudrait encore reprendre intégralement le logiciel, du début jusqu'à la fin, pour ces marchés. Ils préfèrent donc revoir leur stratégie d'ensemble.
- 5 Deux exégèses classiques sont possibles ici. Dans la première, on verra dans le Cas n° 1 un véritable cas de localisation, considérant en quelque sorte que le terme « localisation » devrait être réservé aux seuls cas où il y a clairement adaptation pour un marché donné d'un logiciel dont la conception a été suivant la méthodologie de la globalisation<sup>3</sup>. S'agissant du Cas n° 2, en revanche, on hésitera de parler de localisation, lui préférant peut-être le terme de rétro-conception. La querelle terminologique s'explique par une raison simple qui en dit long sur le sens de plus en plus restreint dans lequel on a aujourd'hui tendance à employer le terme, à savoir que le produit originel n'était pas globalisé, c'est-à-dire qu'il n'était pas conçu avec le marché mondial en vue, avec toutes les conséquences que cela peut avoir pour la conception du logiciel.
- 6 Au sein de la seconde exégèse, on verra dans le Cas n° 1 une approche descendante de la localisation, c'est-à-dire l'éditeur commence en élaborant un produit globalisé pour le marché mondial et puis il procède à la localisation de celui-ci pour des marchés particuliers en adaptant les rares aspects qui peuvent encore varier d'un marché à l'autre. On verra, par contre, dans le Cas n° 2 une approche ascendante de la localisation en ce sens que pour chaque nouveau marché cible, le produit est repris intégralement<sup>4</sup>.
- 7 Pour ma part, j'opterais pour une position proche de la seconde exégèse. Elle consiste à considérer qu'il s'agit dans les deux cas de localisation bien que chacun d'eux représente une étape différente de l'acquisition et du perfectionnement d'un savoir-faire. Les deux cas seront donc approfondis ici. Je note par ailleurs que le Cas n° 2, évacué trop vite par la

première exégèse présente un plus grand intérêt pédagogique ici. Il permet mieux d'attirer l'attention sur tous les aspects de la localisation que le premier pourrait conduire à passer sous silence. Il paraît aussi raisonnable de penser que le Cas n° 2 constitue un plus grand défi pour notre traducteur que le Cas n° 1 pour au moins deux raisons. Premièrement, le concept même de la localisation est à inventer puisque ni le traducteur ni le donneur d'ouvrage n'ont d'idée précise des ramifications du travail qu'ils se proposent de faire et que tous les deux doivent apprendre sur le tas. Deuxièmement, le Cas n° 2 me semble aussi offrir les conditions pour une traduction à forte valeur ajoutée en ce sens que le traducteur doit aussi agir en expert-conseil sur le marché cible. Idéalement, celui-ci doit pouvoir donner des conseils sur les adaptations dont le logiciel a besoin pour augmenter ses chances de réussite dans le marché cible.

## 2. Internationalisation, globalisation, localisation

- 8 On sent déjà l'impossibilité de préciser le sens de la notion de « localisation » sans d'abord préciser celui de l'internationalisation (on dit aussi I18N : I pour la première lettre, 18 pour les dix-huit autres lettres qui interviennent avant la lettre N) et de la globalisation, d'autant que ces deux derniers termes ont souvent été présentés comme allant dans le sens contraire de la localisation<sup>5</sup>. Le premier terme est défini dans le glossaire d'Unicode et de l'internationalisation comme :
- le processus par lequel un logiciel système ou un applicatif est conçu soit pour être indépendant du langage naturel soit pour être transparent à celui-ci (ma traduction).
- 9 Défini de la sorte, on ne voit plus quel sens attribuer à la globalisation sauf à dire que les deux termes sont synonymes, ce qui n'est pas tout à fait le cas. On va donc considérer l'internationalisation comme tout le travail, y compris de théorisation et de normalisation, destiné à créer les conditions permettant de rendre les logiciels et les informations, de façon générale, disponibles à tous les pays, toutes les langues et toutes les cultures. La globalisation, en revanche, est le travail pratique par lequel un produit logiciel est conçu pour être multilingue, typiquement en séparant le noyau invariable du programme de tout ce qui peut varier d'une version nationale à l'autre, en utilisant un jeu de caractères compatible avec le plus de langues possible, en utilisant au maximum des fonctions disponibles au niveau du système d'exploitation, etc.<sup>6</sup> L'objectif recherché en globalisation n'étant pas une affaire de tout ou rien, un produit peut donc être soit entièrement soit partiellement globalisé selon qu'il est compatible avec toutes les langues naturelles connues (ce qui est encore du domaine de l'utopie) ou seulement avec un sous-ensemble de celles-ci.
- 10 Pour en venir à la localisation, on peut la définir comme le processus par lequel un produit conçu pour le marché mondial (produit globalisé) est adapté pour une langue et un pays donnés, étant entendu que :
- l'adaptation en question nécessitera de la traduction mais aussi des conversions diverses permettant un paramétrage socioculturel optimal du produit au marché cible ;
  - l'objectif final est d'aboutir à un produit qui s'utilisera dans le pays cible avec le même naturel que la version internationale.
- 11 On voit donc que même si la globalisation et l'internationalisation sont centripètes dans leurs démarches comparées à la localisation qui est centrifuge, les trois œuvrent en

réalité dans le même sens. L'objectif général recherché par l'internationalisation, à savoir rendre les logiciels et les informations disponibles à toutes les langues et cultures, n'est jamais entièrement atteint sans localisation.

- 12 Il existe aujourd'hui un important savoir-faire en matière de localisation. Malheureusement, ce savoir-faire est détenu et gardé hors de la portée du public par quelques privilégiés (les principaux éditeurs de logiciel et les gros cabinets de localisation) qui considèrent sans doute que la localisation fait partie du cycle du développement des logiciels et, donc, ne relève pas du domaine public. Lorsque les informations sont considérées trop sensibles et de nature à conférer un net avantage technologique sur la concurrence, elles sont gardées secrètes, sinon elles sont partagées au sein de la très fermée « Localization Industry Standards Association », LISA.

### 3. La localisation et la fonction traduction

- 13 Dans le contexte du Cas n° 1, le rôle du traducteur ne s'écartera pas de son rôle habituel. Il s'agira de traduire soit individuellement soit en équipe (avec d'autres traducteurs) la documentation (papier et en ligne) du logiciel ainsi que l'interface utilisateur : les messages d'erreurs, les dialogues, les écrans, etc.
- 14 Dans le contexte du Cas n° 2, et comme je le disais ailleurs, la localisation est à inventer, puisque ni le donneur d'ouvrage ni le traducteur n'ont d'idée claire des ramifications du travail dans lequel ils se sont engagés. Cependant, on peut supposer que la localisation comportera les sous-tâches suivantes : analyse, constitution d'un glossaire, recodage, réécriture de documentation, et test. Or, au sein de ces différentes sous-tâches, le traducteur devenu expert-conseil sur les goûts et les attentes du public du marché cible vis-à-vis du produit sera parfois appelé à assumer des fonctions qui peuvent seulement être de loin assimilables à la traduction.

#### 3.1. Analyse

- 15 Préalable à un travail ordonné, il s'agit ici de dresser la liste des aspects du logiciel méritant considération pour traduction ou conversion en vue de l'adaptation pour le nouveau marché et de proposer des stratégies pour une telle adaptation. Certains de ces aspects sont :
- l'interface (invites et dialogues, messages d'erreurs, écrans d'aide, etc.) ;
  - les paramètres relatifs au traitement des données (la compatibilité clavier, les fonctions de tri, les formats numérique, monétaire, de date, etc., le système de mesure, les dimensions des pages, des enveloppes, etc., les valeurs par défaut, le sens de la saisie de texte, les accents et autres signes diacritiques, la gestion des polices de caractères) ;
  - les manuels utilisateur et de référence, le didacticiel, etc. (les exemples et les exercices d'entraînement proposés dans ceux-ci) ;
  - autres documents (les fichiers d'installation, le fichier d'initialisation, les fichiers « Lisez-moi », les dépliants de présentation).
- 16 Le traducteur peut jouer un rôle capital à ce stade puisqu'il est l'expert sur la culture (au sens le plus large) du marché cible. L'éditeur aura fort besoin de lui non seulement dans le

choix des aspects du logiciel à adapter mais aussi dans le choix des solutions qui passeront mieux dans le marché cible.

- 17 Cette analyse est forcément à caractère coût-avantage puisque l'éditeur est aussi là pour gagner de l'argent. La qualité de logiciel localisé n'étant pas une affaire de tout ou rien, une bonne approche consiste toujours à mettre en balance la localisation de tel ou tel aspect avec des considérations économiques : le public visé, le volume de ventes envisagé, etc. Par exemple, un logiciel technique, destiné à un public spécialisé restreint nécessitera très peu de localisation puisque la documentation sera essentiellement en forme de manuel de référence. En revanche, un logiciel destiné au grand public et qui va forcément se vendre en grande quantité nécessitera une localisation beaucoup plus complète. De la même façon et toujours par rapport au Cas n° 2, il faudra choisir entre l'évolution vers un produit globalisé qui n'aura plus besoin de modifications importantes et dont la rentabilité ne peut être mesurée que dans le long terme ou un produit qui doit être refait à neuf pour chaque nouveau marché.

### 3.2. Constitution du glossaire du projet

- 18 L'idée d'un glossaire constitué préalablement à la traduction à proprement parler (plutôt que tout en traduisant) passe souvent difficilement avec le traducteur. D'une part, c'est un effort qui peut faire double emploi. D'autre part, le traducteur sait que souvent un équivalent qu'on croyait adéquat au début de la traduction d'un long document peut ne plus l'être vers la fin et surtout que les contraintes de mise en discours peuvent parfois amener à ne pas recourir à l'équivalent « officiel ». Malheureusement pour le traducteur rebelle, la pratique est presque la norme en localisation logicielle : très souvent même, le glossaire est livré par le donneur d'ouvrage avec les documents à traduire. Cependant, quand on y réfléchit un peu, le bien-fondé de cette pratique paraît évident. La plupart des projets de localisation mobilisent plusieurs traducteurs qui peuvent même ignorer qu'ils sont plusieurs à travailler sur le même projet. Le glossaire (ainsi que l'éventuel guide de style) constituera donc le gage que le traduit témoignera d'un minimum d'homogénéité de terminologie et de style, non seulement dans la version du logiciel en cours mais aussi à travers les versions successives. Le traducteur peut se consoler aussi du fait que les équivalences proposées dans un tel glossaire ne sont pas gravées sur marbre. Elles peuvent être modifiées à condition que les modifications soient envisagées en consultation avec et au su de tous les partenaires.
- 19 Il convient aussi de souligner — car, on n'y pense jamais assez — que la traduction de la documentation de bon nombre de logiciels nécessitera la maîtrise, en tout cas la facile disponibilité, d'au moins deux terminologies différentes : une terminologie proprement informatique et la terminologie de la spécialité du logiciel. Par exemple, notre traducteur dans le Cas n° 2 aura aussi besoin de la terminologie des transactions commerciales, entre autres.

### 3.3. Réécriture de la documentation pour une meilleure traduisibilité

- 20 Il s'agit de réviser le texte de départ pour en faciliter la traduction. Les éditeurs du calibre et de l'organisation de XYZ dans le Cas n° 1 emploient toujours des rédacteurs professionnels qui se chargent de la rédaction à partir des notes des développeurs de la documentation sur le logiciel. En revanche, chez des éditeurs de la taille de LogiMicro du

Cas n° 2, chaque programmeur est souvent chargé de rédiger la documentation relative à la partie de l'ensemble dont il a la charge de développer. L'expérience montre que le texte est souvent de mauvaise qualité et toute tentative de traduire sans réécriture (même virtuelle) est le moyen le plus sûr d'aboutir à une traduction médiocre. Si l'éditeur ne prévoit pas de faire réécrire la documentation, le traducteur doit envisager de reprendre entièrement le texte dans sa traduction pour mieux refléter ce qu'on sait de la présentation et du fonctionnement du discours technique de façon générale et des spécifications locales pour la langue d'arrivée, des différents sous-genres du discours technique — les textes de procédures, de mises en garde, de description, etc. — et les lois qui les régissent (voir AECMA 1986 et Sides 1992).

### 3.4. Recodage pour faciliter la traduction

- 21 On estime qu'un traducteur traduit une moyenne de 2000 mots par jour de documentation logicielle mais seulement une moyenne de 1500 mots de texte in-code<sup>7</sup>. Donc, le fait d'extraire le texte à traduire du code peut augmenter considérablement la productivité du traducteur. Mais même lorsque ces dialogues sont extraits du code principal pour constituer des fichiers ressource qui sont beaucoup plus souples à gérer, les textes restent encore les arguments des instructions. L'expérience montre qu'il n'est pas toujours certain que le traducteur passe son temps à traduire uniquement ce qui est destiné à l'être. Des cas de codes non destinés à la traduction mais qui finissent par l'être sont bien connus (voir « chaîne » dans l'exemple ci-dessous). Par ailleurs, les textes in-code destinés à l'affichage par le programme sont conçus pour être reconstitués dynamiquement, par exemple<sup>8</sup> :

```
strcpy(texte1, "Le fichier :");  
sprintf(texte2, "%s", chaine);  
strcpy(texte3, " ne peut pas être créé :");  
strcpy(texte4, " blocage réseau. Relancer le logiciel.")
```

- 22 D'abord cette impérative empêche le traducteur de saisir la structure de la phrase à traduire. Ensuite, comme l'ordre des mots de la phrase est presque toujours différent entre langue source et langue d'arrivée, la traduction par sur-écriture qui semble être de rigueur dans de tels cas s'en trouve compliquée<sup>9</sup>. Le traducteur peut donc demander que les textes à traduire soient extraits pour lui faciliter la tâche même si, une fois traduits, de tels textes doivent être réintroduits à leur place.

### 3.5. Localisation des aspects du logiciel qui doivent l'être

- 23 Il s'agit ici de traduire et de convertir tout ce que l'analyse préalable aurait permis de réserver pour traduction ou conversion. Une partie de ce travail (la partie traduction à proprement parler) incombera au traducteur alors que l'autre partie sera le travail de l'éditeur puisqu'elle nécessitera la création de nouveaux codes ou la modification de l'existant selon la stratégie adoptée. Cependant, le traducteur aura la lourde responsabilité d'attirer l'attention de l'éditeur sur de tels aspects. Parce que l'intervalle entre la fin de la localisation et la mise sur le marché du produit est souvent prévu pour être très court, le traducteur visera la production de documents prêts au « flashage ». De la même façon, l'idéal serait que les fichiers ressource arrivent chez l'éditeur comme fichiers de bibliothèques dynamiques prêts à être incorporés dans l'ensemble.

### 3.6. Test de la version localisée

- 24 Le premier test de la version localisée du logiciel revient au traducteur. Il doit suivre le fonctionnement du logiciel étape par étape, écran par écran, pour vérifier que l'aspect des messages écrans et des dialogues est correct (si ceux-ci n'ont pas été programmés pour s'ajuster dynamiquement). Il doit surtout vérifier que la version localisée offre le même niveau de fonctionnalité que la version internationale. Car, la traduction dans ce contexte est à certains égards comparable au doublage et au sous-titrage cinématographiques où ce n'est pas seulement la viabilité linguistique des traductions qui doit en être tenue compte. De même qu'en doublage cinématographique il faut tenir compte de la vraisemblance des traductions proposées par rapport aux mouvements des lèvres, ou qu'en sous-titrage la traduction doit trouver un équilibre entre la durée du plan et le temps nécessaire pour lire les sous-titres, de même en traduisant pour le périphérique d'affichage de l'ordinateur il faut garder constamment à l'esprit que l'espace est donné invariable. C'est donc souvent à la traduction de s'adapter ou, à défaut, au traducteur de re-dimensionner les boîtes de dialogues. Autrement dit, les traductions doivent pouvoir tenir dans l'espace-écran qui leur est alloué.
- 25 À titre anecdotique, je relève cet exemple qui, même s'il ne s'agit pas d'un texte destiné à l'affichage à l'écran, pose le problème avec acuité. Il s'agit d'une instruction #define dont la constante (« Total HT ») doit strictement tenir en dix caractères. Sans précisions particulières, on traduirait en anglais par « Total ex-VAT » ou, à la limite, « Total x-VAT » mais compte tenu des contraintes particulières, le traducteur est obligé d'aller chercher quelque chose comme « Total xVAT ». Mais cela veut dire aussi que celui-ci doit effectivement compter le nombre de caractères de chaque traduction de ce genre, utilisant ici et là le caractère espace pour occuper tout l'espace alloué, ce qui peut peser lourdement sur le rendement de celui-ci.
- 26 Les tests ici sont aussi destinés à s'assurer que le logiciel aura, lors de son utilisation réelle, le comportement que son créateur lui avait prévu et à permettre les captures d'écran utilisées dans la documentation.

## 4. Assurance qualité

- 27 La question de qualité en traduction et de son assurance, surtout lorsqu'il s'agit d'une traduction sous-traitée, sont tributaires de la qualité des intervenants et des contrôles effectués sur le produit aux différents stades. Mais plus généralement, elles dépendent des garde-fous instaurés pour s'assurer du respect des règles élémentaires qui, idéalement, permettraient d'aboutir à un produit à la hauteur des conditions préalablement définies avec le donneur d'ouvrage. Ce sont ces garde-fous que définit la norme ISO sur l'assurance qualité, à savoir l'ISO 9000 de façon générale et, plus particulièrement, le sous-ensemble ISO 9002 qui peut être considéré comme celui qui s'applique à la traduction (GECAP, 1995b). Mais plus concrètement, on peut considérer qu'il y a trois éléments d'appréciation de la qualité d'un produit localisé : la qualité de la traduction, la présentation de celle-ci à l'écran, et le fonctionnement général du logiciel.

## 4.1. La qualité de la traduction

- 28 Donnée pour l'essentiel linguistique (le style, le bon usage, terminologie appropriée), la qualité de la traduction a une influence subjective sur le jugement porté sur le produit localisé. Si un bon logiciel peut l'être en dépit de la mauvaise traduction de sa documentation, on peut penser que celle-ci lui sert de vitrine puisqu'elle est le premier élément d'appréciation pour le public. La traduction doit rechercher l'expression idiomatique, surtout en matière de messages d'erreur. Heureusement qu'il n'y a pas un seul message d'erreur imaginable qui n'ait son « expression consacrée » dans les différentes langues. Par exemple, traduire « *Error: File not found* » par autre chose que « Erreur : Fichier absent », et ce quel que soit le bien-fondé des arguments qu'on pourrait avancer, est comparable à vouloir traduire « *Men at Work* » par autre chose que « Travaux en cours », pour puiser dans les classiques de Vinay et Darbelnet (1977). La même règle s'applique aussi aux noms de menu. L'usage semble avoir consacré, par exemple, le point d'interrogation (?) comme la traduction française du nom de menu « Help » en anglais et « Enregistrer » pour « Save »<sup>10</sup> : le traducteur a intérêt à s'y conformer.

## 4.2. La présentation de la traduction à l'écran

- 29 L'aspect des traductions destinées à l'affichage constitue une autre vitrine du produit localisé. On ne répétera jamais assez la nécessité pour le traducteur d'apporter grand soin à la qualité linguistique des traductions destinées à l'écran. Il faut aussi qu'il apporte autant de soin à la présentation de telles traductions à l'écran et, surtout, qu'il n'oublie pas que l'écran standard est fatalement aux dimensions 80x25, et que la traduction nécessitera très souvent une finition artisanale permettant, par exemple, aux messages affichés de tenir dans les boîtes de dialogue.

## 4.3. Le fonctionnement du logiciel

- 30 La version localisée conserve-t-elle les mêmes fonctionnalités que l'internationale ? Si on peut être tenté d'évacuer cette préoccupation du champ d'action du traducteur dans le Cas n° 1 (puisque'il n'est pas censé s'occuper des autres conversions qui peuvent influencer sur cet aspect du logiciel), cela me paraît plus difficile dans le Cas n° 2 puisqu'il appartiendra souvent à celui-ci d'attirer l'attention sur tout ce qui n'est pas linguistique mais qui peut varier d'un marché, d'une culture à l'autre.

## Conclusion

- 31 En guise de conclusion, je voudrais revenir sur le traducteur — ses qualités personnelles et professionnelles, les outils qui peuvent l'aider, etc. — puisque, en dernière analyse, la réussite de la localisation dépendra en grande partie de son savoir-faire et de l'investissement personnel qu'il apporte à la tâche. Par ailleurs, si le traducteur n'est pas directement tenu de proposer des solutions à certains des problèmes soulevés par la localisation tels que nous les avons évoqués ici, il reste néanmoins vrai que celui-ci peut facilement se trouver au centre d'un projet qui le dépasse, surtout dans le contexte du Cas n° 2. Il y a donc lieu de parler d'une fonction de traduction particulière en localisation qui

fait émerger des exigences spécifiques vis-à-vis du traducteur. Évoquons-en quelques-unes.

- 32 À défaut d'être informaticien, je crois que le traducteur doit être un utilisateur averti des outils informatiques pour au moins deux raisons. D'abord, c'est de cette façon qu'il disposera d'une base de connaissances initiales qui pourrait être complétée par documentation. Ensuite, très souvent celui-ci sera appelé à travailler avec les mêmes outils que le développeur du logiciel s'il tient à proposer des solutions viables dans le temps imparti. J'ajouterai aussi que son expérience d'utilisateur lui sera beaucoup plus utile si elle a été acquise dans le pays cible du logiciel à localiser. Comme je le laissais entendre ailleurs, l'émergence d'un idiome propre aux messages d'erreurs et dialogues oblige le traducteur à doubler son statut de locuteur natif (de la langue d'arrivée de façon générale) de celui d'habitué de cet idiome. Je rappelle que la plupart des cabinets de localisation qui sous-traitent la traduction posent souvent cette condition comme garantie de la connaissance du marché cible par le traducteur.
- 33 Le traducteur doit aussi pouvoir travailler avec des délais serrés et sous pression sans que la qualité du travail n'en souffre, deux exigences *a priori* incompatibles. La partie traduction à proprement parler des projets de localisation est toujours la dernière à démarrer, souvent lorsque le projet n'attend que de telles traductions pour être bouclé. Ceci a pour conséquence que les délais sont souvent courts et, parfois, irréalistes, sans parler des éternelles modifications de dernière minute dans le logiciel qui entraînent nécessairement la modification de la documentation.
- 34 Le traducteur doit aussi se forger une bonne méthode de travail et savoir mettre à profit les outils d'aide à la traduction disponibles aujourd'hui. Après les promesses non tenues des systèmes de traduction automatique, les systèmes dits de mémoire de traduction (par exemple, IBM Translation Manager™ et TRADOS Translator's Workbench™) constituent le meilleur compromis aujourd'hui. Traceurs du processus de la traduction et des solutions proposées pour en faciliter leur rappel, ces outils permettent une maîtrise optimale du processus de la traduction et, donc, du temps psychologique (voir TRADOS 1995a & 1995b)<sup>11</sup>. Le traducteur doit aussi savoir où et comment accéder facilement aux informations qui peuvent l'intéresser dans son travail. La complexification sans cesse croissante des domaines de compétence rend aujourd'hui illusoire tout espoir de bâtir une pratique professionnelle de la traduction sur l'accumulation des connaissances de type factuel, même lorsqu'on a la chance de ne traduire que dans un domaine très restreint. Il faut donc que le traducteur accorde aussi de l'importance aux connaissances procédurales. Si ces dernières ne constituent pas des connaissances directement utilisables sans autres manipulations, elles ont néanmoins l'avantage d'offrir au traducteur les moyens de générer toutes les connaissances factuelles nécessaires à la résolution de tout problème.

---

## BIBLIOGRAPHIE

Abramson, D. 1994. « Developing Windows Software for the global market could be your greatest challenge. Here's how to start ». *Byte*, November, *Globalization of Windows*. <<http://www.byte.com/art/9411/sec9/art7.htm>>.

Adams, G. 1993. « Unicode and internationalization glossary ». <<http://www.stonehand.com/unicode/glosscnt.html>>.

Association européenne des constructeurs de matériel aérospatial, AECMA. 1986. *AECMA Simplified English. A Guide for the Preparation of Aircraft Maintenance Documentation in the International Aerospace Maintenance Language*. AECMA Document PSC-85-16598.

GECAP. 1995a. « Frequently Asked Questions about Localization ». <<http://www.gecap.de/faqs.htm>>

GECAP. 1995b. « The GECAP Quality System ». <<http://www.gecap.de/iso.htm>>.

Sides, C. H. 1992. *How to Write and Present Technical Information*. 2nd Edition. Cambridge : C.ambridge University Press.

Spyropoulos, C. 1993. « GLOSSASOFT Overview & Questionnaire ». <<ftp.vespucci.iquest.com>> <<tatro-enterprises/insoft-l.arc/doc/organizations/glossasoft.txt>>.

TRADOS 1995a. « TRADOS Translator's Workbench for Windows ». <<http://www.mcbsys.com/html/tww.htm>>.

TRADOS 1995b. « Who owns translation memory? ». *MCB News 10*. <<http://www.mcbsys.com/html/news/10/tmowner.htm>>.

VanDevender, S. 1991. « Translation manifesto ». <<ftp.vespucci.iquest.com>> <<tatro-enterprises/insoft-l.arc/doc/guidelines/manifesto>>.

Vinay, J.-P. et J. Darbelnet. 1977. *Stylistique comparée du français et de l'anglais. Méthode de traduction*, nouvelle édition. Paris : Didier.

## NOTES

1. Une première version de cet article a été présentée à la Journée de la Traduction du Groupe de Recherche Multilingue en Traduction Spécialisée (GREMUTS), Université Stendhal - Grenoble 3, 29 mars 1996.
2. Ce point méritait d'être souligné car la facilité incline souvent à penser que l'appellation « version internationale » est réservée à la version américaine. Or, elle désigne en réalité la version originale (*a priori* dans la langue de l'éditeur) à partir de laquelle d'autres versions nationales sont dérivées.
3. «[...] *an approach to designing software that can support the processing of data in all languages simultaneously* »(Abramson, 1994).
4. Sur un plan plus concret, la différence entre les situations décrites dans les deux cas se traduira dans le nombre de modules exécutables que chaque éditeur aura à gérer. Alors que la

société XYZ aura un seul module exécutable pour toutes les versions nationales, LogiMicro aura autant de modules exécutables que de versions nationales (Abramson, 1994).

5. Spyropoulos (1993), par exemple, écrit : « *Localisation is the opposite process [of internationalisation], since it takes a previously internationalised software application adding features and elements that better match the target culture and marketplace.* »

6. Il faut peut-être croire que le terme « globalisation », plus récent par rapport à « internationalisation » et la distinction que nous venons de faire ont plus cours outre-Atlantique, car en Europe seul le dernier terme et ses formes dérivées sont employés.

7. Par texte in-code, il faut entendre un texte (normalement destiné à l'affichage) enfoui dans les instructions qui le manipulent.

8. Dans cet exemple, le programme doit récupérer le nom du fichier dont la création pose problème et procéder à la concaténation des différents morceaux de la phrase pour pouvoir afficher un message comme « Le fichier monfich.tdr ne peut pas être créé ».

9. Dans son manifeste de la traduction, qui est en réalité des propositions pour la conception de programmes qui prévoient d'éventuelles localisations pour d'autres marchés, VanDevender (1993) s'étend longuement sur les problèmes occasionnés par ce mélange de code et de texte en localisation et comment les éviter. Ces solutions relèvent toutefois, comme on pouvait s'y attendre, du domaine d'intervention du programmeur, ce qui souligne le dilemme du traducteur qui n'a pas pris sur tous les facteurs qui influent sur la qualité du travail qu'il est appelé à faire.

10. Il fut un temps où, par exemple, « Sauvegarder » était en concurrence avec « Enregistrer » comme traduction de l'anglais « Save » et « Erreur : Fichier introuvable » en concurrence avec « Erreur : Fichier absent » pour traduire « Error: File not found ». De toute évidence, aujourd'hui la dernière de chaque paire semble l'avoir emporté définitivement.

11. Le temps psychologique est entendu ici au sens du sentiment du sujet de la commune mesure entre une tâche donnée et le temps physique passé à l'accomplir. Alors que la mesure du temps physique s'exprime en termes de nombre d'heures, la mesure du temps psychologique s'exprime dans des termes subjectifs comme : très peu de temps, plus de temps qu'il n'en faut, juste le temps qu'il faut, etc. Deux traducteurs peuvent passer le même nombre d'heures sur une tâche sans qu'il ne s'agisse du même temps psychologique. Mon sentiment est que ce qui est important du point de vue de la motivation et l'éveil du traducteur pour la tâche en cours est la maîtrise de ce temps psychologique et que tout ce qui peut retarder le jugement de disproportion entre une tâche et le temps passé à l'accomplir œuvre dans le sens de l'amélioration de la qualité du travail du traducteur.

---

## RÉSUMÉS

Adaptation d'un logiciel aux goûts, normes et culture du peuple d'un marché étranger, la localisation logicielle implique forcément de la traduction mais elle ne s'y réduit pas. Du fait du dédoublement des difficultés habituelles liées à la traduction de façon générale des contraintes spécifiques à la localisation, la fonction traduction peut y relever d'un exercice périlleux, surtout si le donneur d'ouvrage lui-même n'a pas beaucoup d'expérience en la matière. En s'appuyant sur deux cas fictifs, cet article propose une discussion générale sur la spécificité de la fonction traduction dans ce contexte et comment aborder certains des problèmes soulevés par l'exercice.

Insofar as software localization has to do with the adaptation of a software package to the taste, standards, and culture of other peoples and markets, it necessarily involves translating but it cannot by any stretch of the imagination be subsumed under translation. The usual difficulties involved in any form of translation combine here with localization-specific constraints to make the translator's role in a software localization project especially challenging, in particular if the client himself has no localization experience. Working from two fictitious case studies, this article presents a general discussion on the specificity of translation in this setting and how to handle some of the problems posed.

## INDEX

**Mots-clés** : globalisation, internationalisation, localisation logicielle, traduction

**Keywords** : globalization, internationalization, software localization, translation

## AUTEUR

### UZOMA CHUKWU

Uzoma Chukwu a soutenu en 1993 une thèse de doctorat de l'Université Lumière - Lyon 2 en études anglophones sur le « Repérage des termes dans un corpus bilingue anglais/français ». Membre du Centre de recherche en terminologie et traduction (CRTT), Université Lyon 2, il est actuellement traducteur/terminologue indépendant et maître de conférences associé au Service d'enseignement des langues (SEL) de l'université Claude Bernard - Lyon 1.  
uchukwu@gowebway.com