Traduire
Revue française de la traduction

224 | 2011
Des outils et des nuages

Les outils libres du traducteur, un écosystème à apprivoiser

Didier Briel

Édition électronique
URL : http://journals.openedition.org/traduire/184
DOI : 10.4000/traduire.184
ISSN : 2272-9992

Éditeur
Société française des traducteurs

Édition imprimée
Date de publication : 1 juin 2011
Pagination : 38-49
ISSN : 0395-773X

Référence électronique
Didier Briel, « Les outils libres du traducteur, un écosystème à apprivoiser », *Traduire* [En ligne], 224 | 2011, mis en ligne le 10 février 2014, consulté le 30 avril 2019. URL : http://journals.openedition.org/traduire/184 ; DOI : 10.4000/traduire.184

Les outils libres du traducteur, un écosystème à apprivoiser

Didier Briel

Pour nombre de traducteurs, les outils de Traduction Assistée par Ordinateur font maintenant partie du quotidien, au même titre que le traitement de texte et le navigateur web. Évoquer l'existence de logiciels libres dans cette catégorie suscite fréquemment des questions sur la nécessité d'utiliser Linux, ou sur la gratuité. Cet article tente de répondre à ces questions, et présente quelques-uns de ces logiciels : OmegaT, logiciel de TAO, Okapi, un ensemble de composants dédiés à la localisation et à la traduction, et bitext2tmx, bligner et LF Aligner, trois logiciels d'alignement.

Libertés fondamentales

Il y a souvent confusion entre *freeware* et logiciel libre, en partie à cause de la ressemblance des dénominations en anglais (*freeware* et *free software*). Disons-le sans ambiguïté, la liberté n'a que peu à voir avec la gratuité, puisqu'il est même possible de vendre des logiciels libres.

Ce qui caractérise un logiciel libre, ce sont les quatre libertés fondamentales qu'il octroie à ses utilisateurs.

Vous devez pouvoir exécuter le programme, pour n'importe quel usage. Si un logiciel vous interdit de l'utiliser dans un cas particulier (ex. : pour la recherche dans un domaine controversé), il ne s'agit pas d'un logiciel libre.

Vous devez pouvoir étudier le fonctionnement du programme, et l'adapter à vos besoins. Cela implique que l'accès au code source du logiciel soit garanti. Même si vous ne programmez pas, cette liberté est très importante. Cela permet la transparence absolue sur ce que fait, ou ne fait pas, un logiciel.

Par exemple, les logiciels de TAO récents comportent souvent une possibilité de traduction automatique basée sur Google Translate. À ce titre, une question revient fréquemment : est-ce que le logiciel n'envoie pas ma traduction modifiée à Google Translate, afin que celui-ci

s'améliore ? Dans le cas d'un logiciel « propriétaire » (un logiciel non libre), vous devez croire sur parole l'auteur ou l'éditeur du logiciel s'il vous affirme que non. Dans le cas d'un logiciel libre, chacun peut constater (ou faire constater par une tierce partie indépendante) que la traduction n'est jamais renvoyée à Google.

D'autre part, adapter un logiciel libre est parfois trivial, et accessible à des personnes sans connaissances spécifiques en programmation. Si vous ne pouvez pas le faire vous-même, vous pouvez toujours demander à quelqu'un de le faire pour vous.

Vous devez pouvoir redistribuer des copies du logiciel, à qui vous voulez. Au lieu de renoncer à un travail en commun parce que votre interlocuteur ne dispose pas de votre logiciel, vous pouvez lui en donner un exemplaire. Personne ne vous empêche d'ailleurs de vendre votre copie. Évidemment, si vous le faites dans les mêmes conditions que ceux qui la distribuent gratuitement, il est peu probable que vous ayez beaucoup d'amateurs, mais cette vente pourrait correspondre à des services : installation du logiciel sur support physique, manuel imprimé, etc.

Enfin, vous devez pouvoir distribuer à d'autres des copies de vos versions modifiées, afin d'en faire bénéficier la communauté. Cela aussi implique l'accès au code source du logiciel. Si vous effectuez, ou faites effectuer des adaptations, c'est une garantie de pérennité. Vos changements étant publics, ils peuvent être repris par d'autres, et servir de base pour de nouvelles améliorations.

Écosystème des outils libres de TAO

En dehors de Linux, qui est un système d'exploitation, les logiciels libres les plus connus des traducteurs sont sans doute Firefox, le navigateur web, et OpenOffice.org, existant maintenant également sous la forme LibreOffice, la suite bureautique concurrente de Microsoft Office.

Ces applications ont en commun le fait d'être multiplateformes, c'est-à-dire de pouvoir s'exécuter sous Windows, Linux et Mac OS X.

Cette capacité d'exécution sous différents systèmes permet de fournir des outils à des systèmes moins bien pourvus, ou d'augmenter considérablement la part de marché, lorsque le logiciel n'était pas initialement disponible sous Windows. C'est techniquement réalisable en ne recourant pas à des méthodes ne marchant que sur un seul système, et éventuellement en adaptant une partie du logiciel pour chaque système.

Les outils libres pour traducteurs étant d'une diffusion incomparablement plus restreinte, ils ont en commun le fait d'être multiplateforme à moindre frais, c'est-à-dire de limiter considérablement l'adaptation aux différents systèmes : en s'appuyant sur une machine virtuelle ou un interpréteur, le programme fonctionnera sur toute machine disposant de la machine virtuelle ou de l'interpréteur.

Cela explique pourquoi la majorité de ces outils fonctionnent sous la machine virtuelle Java (OmegaT, Okapi et bitext2tmx), les autres s'appuyant principalement sur un langage interprété (Perl ou Python). Tous les logiciels cités dans cet article fonctionnent donc indifféremment sous Windows, Linux et Mac OS X, avec parfois quelques différences mineures selon la plateforme.

À la fois pour des raisons d'économie et de « philosophie » (aussi bien technique que fonctionnelle), les outils libres ont beaucoup moins tendance à être des outils « tout en un ». Il paraît par exemple peu rationnel de créer un assistant complexe pour effectuer des tâches triviales (ex. : copier des fichiers), alors que le système d'exploitation hôte dispose d'outils sophistiqués pour le faire. D'autre part, il n'y a pas d'impératif commercial poussant à fournir « autant de fonctions que le concurrent ». Si un autre logiciel libre fournit une fonction de façon satisfaisante, il ne sert à rien de la redévelopper.

Les mêmes motifs poussent les logiciels libres à s'appuyer sur des formats standards et ouverts. Le code source étant disponible, il n'y a aucun intérêt à s'appuyer sur des « secrets de fabrication ». Au contraire, l'utilisation de standards permet de réduire les coûts de développement, et d'augmenter l'interopérabilité : ce qui ne peut pas être fait dans un logiciel pourra l'être dans un autre. Les données sont ainsi rarement stockées de façon binaire dans les logiciels en question ; à la transparence du code source, s'ajoute la transparence des données, lesquelles peuvent presque toujours être examinées à l'aide d'un simple éditeur de texte.

Le standard commun à tous ces logiciels est TMX, le standard d'échange des mémoires de traduction. Sans garantir systématiquement une interopérabilité parfaite, il permet néanmoins un flux d'échanges largement utilisable. Les standards de fait que sont les fichiers texte, où les données sont séparées par des virgules ou des tabulations (CSV ou TSV), sont également largement utilisés. S'y ajoutent XLIFF, un format d'échange de fichiers multilingues, TBX, le format de base terminologique (normalisé sous la dénomination ISO 30042), SRX, la description de règles de segmentation, et ITS, un langage de description des éléments à traduire.

Par essence, les logiciels libres ne sont pas des entités autonomes et isolées, mais interdépendantes. Ce qu'on appelle un programme s'appuie souvent en partie sur d'autres programmes ou « bibliothèques ». OmegaT utilise ainsi le vérificateur orthographique Hunspell (celui utilisé par Firefox, OpenOffice.org et LibreOffice), le vérificateur linguistique LanguageTool et les lemmatiseurs de Lucene.

Les relations entre applications peuvent même devenir symbiotiques. Okapi dispose d'options dédiées à l'utilisation d'OmegaT, et OmegaT a été modifié pour collaborer plus efficacement avec Okapi.

LF Aligner est une autre illustration de la réutilisation maximale permise aux logiciels libres : l'application n'effectue pratiquement aucune tâche par elle-même, agrégeant non moins de sept logiciels différents (y compris un traitement de texte) pour parvenir à ses fins.

Au titre de la réutilisation, on notera qu'OmegaT est à l'origine d'autres projets dont il est le moteur, par exemple Boltran, un « OmegaT *anywhere* », c'est-à-dire une version HTML utilisable depuis un simple navigateur, et Autshumato ITE, un intégré dédié aux onze langues officielles d'Afrique du Sud, qui incorpore OpenOffice.org et OmegaT dans une même application. La réutilisation n'est pas à sens unique, puisque ces deux projets ont apporté en retour certains de leurs développements à OmegaT.

Répartition des rôles

À la lumière de ce qui a été exposé précédemment, il ne faut pas considérer chaque application comme un tout isolé, au périmètre limité, mais comme l'un des éléments d'une trousse à outils modulable en fonction des besoins et des préférences.

La partie « mémoire de traduction », et plus généralement « traduction assistée par ordinateur » est tenue par **OmegaT**. Il gère les éléments principaux de ce type d'application : saisie des traductions, correspondances partielles, concordance, utilisation de glossaires.

Okapi est la boîte à outils de la trousse à outils. Il s'agit d'un ensemble de fonctions dédiées à la localisation et à la traduction. Ces fonctions sont regroupées dans deux programmes principaux : Rainbow, chargé principalement des conversions entre différents formats de documents et Checkmate, dédié à l'assurance qualité.

Rainbow sert principalement de préparateur de projet pour OmegaT. En utilisant ses 28 filtres de formats de fichiers, il est possible d'étendre de façon significative le nombre de formats de documents qu'il est possible de traduire avec OmegaT. Des options dédiées permettent de créer automatiquement un projet OmegaT complet, avec récupération automatique des éventuelles traductions existantes dans les documents source. Une fois le projet traduit, les documents cible sont recréés automatiquement dans le format original. Utiliser Rainbow plus OmegaT revient donc à procéder de la même façon que nombre de logiciels commerciaux de TAO : conversion des fichiers source dans un format intermédiaire, traduction, reconversion du format intermédiaire dans le format original.

Rainbow permet également toutes sortes de traitements annexes et pratiques : recherche et remplacement, extraction terminologique, conversion d'un format bilingue vers un glossaire tabulé, etc.

Enfin, certains des filtres de Rainbow sont directement disponibles dans OmegaT sous la forme d'un module d'extension.

Checkmate est une application d'« assurance qualité ». Elle permet toutes sortes de vérifications automatiques : mots répétés, espaces superflues ou manquantes, « modèle » différent entre

la source et la cible (ex. : nombre de parenthèses différent), ainsi, en version bêta, qu'une vérification terminologique sur la base d'un glossaire. Checkmate peut être utilisée de façon autonome sur tout document bilingue (ex. : TMX, RTF bilingue) mais également en temps réel en conjonction avec OmegaT : à chaque sauvegarde de la mémoire, ou à chaque création des documents cible, toutes les vérifications sélectionnées peuvent être effectuées à nouveau de façon automatique.

Stratégies d'alignement

Pour compléter le tableau, il ne manque plus que de pouvoir récupérer des traductions existantes à partir de documents monolingues déjà traduits.

OmegaT le permet pour quelques formats de localisation (PO et *properties* Java), et Rainbow possède quelques fonctions dédiées à cet effet, mais qui ne fonctionnent bien que pour des textes parfaitement alignés au préalable.

On aura donc intérêt à étudier d'autres possibilités. En matière d'alignement, il existe deux grandes stratégies.

On peut effectuer un alignement « visuel », c'est-à-dire que le logiciel présente un tableau dans lequel il a essayé, avec plus ou moins de bonheur, d'apparier les segments source et cible. Il reste ensuite à l'utilisateur à corriger l'alignement en fusionnant ou en découpant des segments. Cela peut être fastidieux, mais le résultat est par définition parfait, puisque c'est l'utilisateur qui le définit. Ce choix est généralement judicieux pour des volumes pas trop importants, et lorsqu'une fiabilité totale est requise.

Ou on peut laisser travailler le logiciel de façon automatique, quitte à passer un certain temps à ajuster les options et à préparer les documents source et cible, et quitte également à accepter une certaine imprécision. Si, par exemple, on doit aligner un *corpus* de 300 000 mots, qui serviront essentiellement de référence, peu importe si une dizaine de phrases se retrouvent sans correspondance.

Pour l'alignement visuel, la seule solution libre est **bitext2tmx**. Il faut commencer par convertir les documents à aligner en fichiers texte, ce qui se fait généralement facilement par un « Enregistrer sous » dans l'application d'origine. Le programme est ensuite très simple à appréhender et, après quelques tâtonnements (« fusionner » s'applique-t-il au segment suivant ou précédent ?) on trouve assez rapidement ses marques. Son principal défaut est que les règles de segmentation préalables sont figées, et ne sont pas modifiables par l'utilisateur. En particulier, il n'est pas possible d'introduire des exceptions pour ne pas segmenter sur les abréviations usuelles. Si cela ne pose en général pas trop de problèmes pour de nombreux textes en anglais ou en français, où une correction manuelle a vite fait de réparer les quelques

dérapages, il en va autrement d'un texte foisonnant d'abréviations, comme cela arrive par exemple fréquemment en allemand, où le nombre de segments à corriger peut vite devenir dissuasif.

Pour l'alignement automatique, il existe beaucoup plus de solutions. Nous en avons retenu deux, présentant deux aspects opposés.

Bligner est une application spartiate, puisque tout se résume au code source, qui sert à la fois à l'exécution du programme et à sa configuration. Ce code source est disponible en Python et en Perl, selon les préférences de chacun. Il est à noter que si, sous Windows, installer l'un ou l'autre de ces interpréteurs nécessite un (petit) effort de la part de l'utilisateur (il existe des programmes d'installation « tout-en-un »), les interpréteurs sont déjà présents dans toute distribution Linux et tout système Mac OS X.

Comme bitext2tmx, bligner fonctionne à partir de fichiers texte. Contrairement à bitext2tmx, les règles de segmentation sont entièrement configurables, dans une syntaxe proche de celle utilisée par Okapi et OmegaT, qui est celle du standard SRX. Bligner peut également fonctionner en mode paragraphe, la fin de paragraphe étant alors la seule marque de changement de segment.

Dans tous les cas, l'alignement ne sera possible que si le nombre de paragraphes source et cible est strictement identique. C'est là l'inconvénient principal du logiciel. En cas de décalage, il faudra donc corriger l'un ou l'autre fichier (avec un simple éditeur de texte) afin de rectifier la situation.

Par contre, bligner gère sans difficulté une disparité dans le nombre de phrases à l'intérieur d'un paragraphe. Il utilise un algorithme simple(t) pour reporter les phrases superflues à la fin, en les renumérotant. Ce nombre supplémentaire sera suffisant pour qu'un logiciel de TAO n'identifie pas cette traduction comme une correspondance exacte. On en revient à la finalité de l'opération : dans un alignement exécuté « pour référence », peu importe le décalage de quelques phrases, puisque de toutes façons un traducteur (humain) examinera à nouveau chaque proposition de traduction. Bligner peut fonctionner en mode « interactif » (les documents à aligner, ainsi que les langues source et cible, peuvent être saisis directement), mais fonctionne également en ligne de commandes, ce qui permet une automatisation complète lors du traitement d'un grand *corpus*. Au final, le plus grand défaut de bligner est également sa plus grande qualité : son extrême simplicité.

Avec une même finalité d'alignement automatique, **LF Aligner** adopte une approche radicalement différente : intelligence, nombreuses possibilités et complexité sont au rendez-vous. À partir de l'application principale (qui est développée également en Perl, mais livrée compilée sous Windows), l'utilisateur saisit un certain nombre de paramètres, puis laisse la magie s'opérer. Le résultat est souvent très bon, à la limite de la magie noire, comme le remarque l'auteur. C'est

que LF Aligner s'appuie sur Hunalign, doté d'un puissant algorithme de rapprochement. Il est possible d'améliorer encore ces résultats avec des dictionnaires spécifiques à des combinaisons linguistiques. Quelques-uns sont fournis, comme anglais-hongrois ou italien-anglais. Mais, même sans cela, on obtient d'excellents résultats. À une étape du processus, il est possible d'utiliser une approche semi-visuelle : l'alignement s'affiche dans Excel (sous Windows), et il est possible de corriger des décalages avant la validation finale.

En plus des fichiers texte, LF Aligner permet de travailler à partir de plusieurs formats d'entrée : .doc, .docx, .rtf et .pdf. Pour chacun de ces cas, il appelle en fait une application qui va se charger de convertir le fichier en question en fichier texte. De plus, LF Aligner permet également d'aligner des *corpus* bilingues directement depuis Internet, notamment la législation européenne.

Les qualités de LF Aligner sont évidentes. Une fois apprivoisée, la productivité obtenue devrait être bien supérieure à celle offerte par les outils classiques des logiciels commerciaux de TAO. Ses défauts : une certaine complexité, mais un *howto.txt* permet de s'y retrouver pour une première approche simplifiée, et des conversions en fichier texte qui remplissent les segments de caractères parasites, par exemple, les puces systématiquement conservées sous forme d'astérisques. Dans la mesure du possible, on aura donc intérêt à effectuer soi-même la conversion en fichiers texte.

Les méthodes présentées ci-dessus ne doivent pas faire oublier que, grâce à des standards tous simples (fichiers texte et TMX), il est possible de mélanger différentes approches. Samuel Murray a ainsi conçu une méthode intéressante d'alignement, décrite dans un tutoriel vidéo(1). Le texte est tout d'abord segmenté dans OmegaT, transformé en fichier texte par simple copier/coller, puis aligné avec bligner.

OmegaT, un outil professionnel

Ce qui caractérise le plus OmegaT, c'est qu'il s'agit d'un outil conçu par, et pour, des traducteurs professionnels. Le coordinateur du projet, Marc Prior, est traducteur, et non informaticien, et ceux-ci sont en minorité parmi les « administrateurs » (personnes ayant des responsabilités au sein du projet OmegaT).

Cet aspect professionnel est renforcé par le fait que, contrairement à ce qu'on pourrait croire, OmegaT n'est pas un outil très utilisé par les « traducteurs du libre ». Il n'est pas forcément adapté à leurs procédures, et il n'est définitivement pas un outil « pour informaticien ».

(1) http://www.bligner.org/francais/printable/documentation/tutoriel-video/tutoriel-video.html

En ce qui concerne le fonctionnement général, OmegaT est dans la catégorie des logiciels autonomes. Il dispose de sa propre interface utilisateur, et rejoint en cela la tendance actuelle adoptée par la majorité des acteurs du marché. À ce titre, la seule différence notable est qu'il affiche les segments source et cible l'un au-dessus de l'autre plutôt que côte à côte. Cet affichage « en ligne » a l'avantage de permettre un mode simulant le mode « par écrasement », et facilitant une relecture contextuelle. On peut choisir de ne voir que les segments cible pour les segments déjà traduits, et que les segments source pour ceux restant à traduire.

Contrairement aux tendances actuelles, par contre, l'interface utilisateur est d'un dépouillement extrême : le logiciel ne comporte pratiquement aucune icône, mises à part celles servant à la configuration des volets.

OmegaT fonctionne sur le principe du projet : on n'ouvre pas un document, on traduit un projet qui contient un ou plusieurs documents. Le nombre de documents source peut être grand (il n'y a pas de limite) et comporter un nombre quelconque de dossiers. Fait très utile dans les projets complexes, l'arborescence de la source est reproduite à l'identique dans la cible, y compris les fichiers n'étant pas à traduire (images, etc.). Cela s'avère particulièrement pratique pour traduire des sites web ou des aides en ligne, parfois composés de milliers de fichiers.

Le nombre de formats reconnus en entrée est assez important (24, 29 si l'on compte ceux fournis par le module d'extension d'Okapi), avec un nombre important de formats de localisation. Les formats Microsoft Office propriétaires (« MS Office 97-2003 ») sont notoirement absents. Si cela a longtemps été un frein à l'adoption par certains utilisateurs, l'utilisation croissante du format Open XML (« MS Office 2007-2010 »), qui est lui pleinement pris en charge, enlève beaucoup d'importance à cette lacune. Même si c'est au prix de quelques restrictions, les formats Trados TTX et SDLXLIFF sont également pris en charge, ce qui enlève une autre bonne raison de ne pas utiliser OmegaT.

Techniquement, OmegaT est basé sur des choix originaux, qui sont à la fois des avantages et des inconvénients.

Première originalité, OmegaT n'utilise aucun format intermédiaire. Les fichiers source sont analysés, leur contenu est extrait en mémoire, et le fichier cible est réécrit directement en substituant la traduction au texte original. Le logiciel y gagne une agilité certaine, les phases de préparation (en fait, de conversion) d'un projet ainsi que les phases de « nettoyage » étant tout simplement absentes. La modification du contenu d'un projet (rajouter ou supprimer des fichiers) peut se faire à la volée, et ne prend que le temps de la copie physique des fichiers dans le bon dossier. D'autre part, le formatage du document est très rarement impacté, puisque aucune conversion n'a eu lieu.

Deuxième originalité, OmegaT ne s'appuie pas sur une base de données, mais travaille entièrement en mémoire. Conséquence : tout ou presque est « instantané » (dans OmegaT, on

compte généralement en millisecondes, pas en secondes), qu'il s'agisse de la recherche (« concordance » dans d'autres logiciels) ou du calcul des correspondances partielles. Par rapport à la lourdeur de certains autres produits, la différence est significative, et fait partie du professionnalisme de la solution. Inconvénient : la mémoire n'étant pas illimitée, le nombre d'unités de traduction (de segments) qu'il est possible de gérer dans un projet ne l'est pas non plus. Cela est particulièrement vrai sous Windows 32 bits, où la mémoire utilisable ne dépasse pas 3,3 Go. En pratique, quelles sont les limites ? Un projet de 300 000 mots se gère sans aucun problème. Par contre, il n'est pas possible de charger la mémoire de traduction anglais-français de la Direction Générale de la Traduction.

Troisième point : l'application de traductions existantes dans un nouveau document et la propagation bidirectionnelle des répétitions sont immédiates et instantanées. Tout segment identique à un segment modifié reflète instantanément la modification. Les avantages évidents sont la vitesse d'exécution, et surtout la cohérence : dans des manuels particulièrement répétitifs, le gain est très significatif. Inconvénient, il n'est pas possible d'avoir deux traductions différentes pour un même segment source. C'est plus gênant en localisation (il y a souvent des phrases très courtes, ce qui augmente les possibilités de traductions différentes) qu'en traduction, et encore plus lorsqu'on traduit vers des langues comportant de nombreuses flexions. Même si bien sûr il y a des contournements, aucun n'est parfaitement satisfaisant, et c'est un nouveau point d'achoppement pour certains. Une version résolvant définitivement ce problème est déjà développée, et en est au stade de la version alpha.

Lorsqu'on est habitué à un logiciel, il est facile de ne voir dans les autres que ce qui est « moins bien », et de déclarer immédiatement qu'ils ne sont pas utilisables. Et pourtant, nombre de traducteurs professionnels utilisent OmegaT, même s'il n'est pas vraiment possible d'avoir des chiffres d'utilisation fiables pour un logiciel libre tel qu'OmegaT (les seul chiffres disponibles étant le nombre d'inscrits à la liste OmegaT sur Yahoo : 1 518 au 30 mars 2011, et le nombre de téléchargements mensuels, qui se situe généralement entre 4 000 et 5 500).

C'est qu'OmegaT n'a pas que des défauts et, en plus des points cités ci-dessus (rapidité, simplicité et agilité), qui en eux-mêmes peuvent déjà susciter l'intérêt, comporte quelques particularités non négligeables. Par exemple, pour la gestion, un nombre illimité de documents source, de mémoires de traduction et de glossaires ou, du point de vue algorithmique, la prise en compte automatique des flexions dans les glossaires et dans la recherche des correspondances partielles, grâce aux lemmatiseurs de Lucene. Il faut mentionner également la prise en charge des dictionnaires aux formats StarDict et Lingvo DSL.

La traduction automatique par Apertium, en plus de celle de Google Translate, une excellente gestion des combinaisons linguistiques incluant une langue écrite de droite à gauche, ainsi qu'un vérificateur linguistique (LanguageTool) comportant des règles dédiées (même peu nombreuses) à des langues telles que le galicien et l'islandais ne manquent pas d'intéresser les traducteurs ne travaillant pas dans les combinaisons linguistiques les plus courantes.

Il faut dire que le multilinguisme est ancré dans la culture d'OmegaT. L'interface utilisateur est disponible dans 27 langues (même si toutes ne sont pas à jour), et une documentation récente et complète est disponible dans neuf langues.

Les logiciels libres, un modèle différent d'investissement

Contrairement à certains logiciels libres à destination des entreprises, OmegaT n'est pas un logiciel commercial, et ne dépend d'aucune société. Il n'existe pas de version payante, et c'est là son plus grand défaut : comment prendre au sérieux un outil entièrement gratuit ? Même si les qualités de Firefox, ou l'utilisation gratuite des ressources d'Internet, ont commencé à faire bouger certaines mentalités, l'idée que le professionnalisme passe par un investissement conséquent reste souvent présente.

Les deux raisons les plus souvent évoquées pour ne pas utiliser un logiciel sont l'absence de support, et l'absence de telle ou telle fonctionnalité, qui est considérée comme bloquante.

Le support est fourni par les utilisateurs sur un groupe Yahoo, et la couverture temporelle et linguistique pourrait faire l'envie de certaines sociétés commerciales, l'assistance pouvant s'effectuer au minimum en japonais, russe, allemand, français et anglais, depuis le matin au Japon jusqu'en soirée en Amérique.

Malgré cela il est évident que, dans certains cas d'utilisation, une relation contractuelle payante sera plus rassurante pour un professionnel. Rien n'empêche alors de souscrire un tel contrat d'assistance avec une entreprise ou un indépendant. Le prix d'une licence de base de Trados (845 € au 30 mars 2011), par exemple, permet d'acheter de nombreuses heures d'assistance. Une autre solution, pour des sociétés disposant de ressources informatiques, est de former l'un des informaticiens de l'entreprise au dépannage et même au développement d'OmegaT. Certaines améliorations du produit proviennent ainsi de développements internes d'un éditeur de logiciels médicaux.

L'absence de telle ou telle fonctionnalité est souvent mise en avant pour justifier l'impossibilité d'utiliser un logiciel donné. Lorsqu'il s'agit d'un logiciel propriétaire, l'argument est en effet définitif. Dans le cas d'un logiciel libre, il est possible d'envisager les choses autrement.

Faire effectuer un développement peut permettre d'obtenir une adaptation fine de la solution aux besoins, pour un coût parfois inférieur à celui d'une seule licence propriétaire. Dans l'équation économique, il faut, de plus, non seulement prendre en compte le coût initial du logiciel, mais également le coût de gestion des licences correspondantes. Le logiciel libre peut en effet être déployé sur un nombre illimité de postes, sans aucune gestion de licence. Même pour un indépendant, il n'est pas rare de disposer de trois machines (une machine de bureau, un portable et un *netbook*). Il faut alors jongler avec la ou les licences flottantes, en espérant que si une machine connaît une défaillance, ce ne sera pas justement celle qui détient la licence en activité.

Au contraire, un logiciel libre tel qu'OmegaT peut être immédiatement disponible, non seulement sur toutes les machines de l'utilisateur, mais également installé à demeure de façon « portable » sur une clé USB, ou même en version sans installation (Java Web Start), utilisable depuis n'importe quelle machine connectée à Internet.

La pérennité de l'investissement est également à prendre en compte. Sauf cas particulier, les développements effectués sur des logiciels libres seront rendus publics, et intégrés dans la version standard. Ils seront donc maintenus, gratuitement, par les développeurs des versions futures.

Au final, le coût d'achat d'un logiciel (et donc son éventuelle gratuité) n'est qu'un des paramètres à prendre en compte pour évaluer le coût d'usage, et le professionnalisme d'une solution ne peut se mesurer au coût de la licence.

L'investissement intellectuel et sa rentabilité ne sont bien sûr pas à négliger. Par leur simplicité et leur absence de sophistication, les outils décrits dans cet article demandent un effort intellectuel différent (pas forcément plus « difficile ») de celui exigé par certains logiciels commerciaux. En échange, on y gagne une compréhension des processus plus profonde que celle offerte par des successions de clics et des certifications basées sur des questionnaires à choix multiples.

didier@didierbriel.fr

*Après une orientation initiale littéraire, **Didier BRIEL** a effectué une première carrière dans l'informatique, notamment comme architecte produit, directeur technique et directeur marketing et stratégie au sein d'éditeurs européens de logiciels pour entreprises. Il est traducteur anglais-français depuis 2004, et est membre de la SFT. Il a commencé à contribuer au logiciel OmegaT en 2006, et en est devenu responsable du développement en 2007. Il est par ailleurs l'auteur du script d'alignement bligner.*

Références

Projet OmegaT, 2011, *OmegaT*, http://www.omegat.org/, consulté le 30 mars 2011.

Okapi, 2011, *Okapi Framework*, http://okapi.opentag.com/, consulté le 30 mars 2011.

bitext2tmx, 2011, *bitext2tmx: Bitext Aligner/Converter,* http://bitext2tmx.sourceforge.net/, consulté le 30 mars 2011.

BRIEL Didier, 2011, *Aligneur bligner Perl et Python*, http://www.bligner.org/francais/, consulté le 30 mars 2011.

Farkasandras, 2011, *LF Aligner,* https://sourceforge.net/projects/aligner/, consulté le 30 mars 2011.

Free Software Foundation, Inc., traduction Karl Pradène, 2010, *Définition d'un logiciel libre,* http://www.gnu.org/philosophy/free-sw.fr.html, consulté le 30 mars 2011.

LISA, 2005, *TMX 1.4b Specification*, http://www.lisa.org/Translation-Memory-e.34.0.html, consulté le 30 mars 2011.

OASIS, 2008, *XLIFF Version 1.2*, http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html, consulté le 30 mars 2011.

LISA, 2008, *Systems to manage terminology, knowledge, and content – TermBase. eXchange (TBX)*, http://www.lisa.org/fileadmin/standards/TBX_2008_10_29.pdf, consulté le 30 mars 2011.

LISA, 2008, *SRX 2.0 Specification*, http://www.lisa.org/fileadmin/standards/srx20.html, consulté le 30 mars 2011.

W3C, 2007, *Internationalization Tag Set (ITS) Version 1.0,* http://www.w3.org/TR/its/, consulté le 30 mars 2011.

ByteTranslation, 2010, *Boltran*, http://www.boltran.com/, consulté le 30 mars 2011.

North-West University, 2009, *Autshumato*, http://autshumato.sourceforge.net/, consulté le 30 mars 2011.

MURRAY Samuel, 2010, *Tutoriel vidéo sur l'alignement*, http://www.bligner.org/francais/documentation/tutoriel-video/tutoriel-video.html, consulté le 30 mars 2011.

GARCÍA-VERDUGO José, 2010, *OmegaT at the University (Arabic-Spanish translation)*, http://tech.groups.yahoo.com/group/OmegaT/message/19344, consulté le 30 mars 2011.