

L^AT_EX, un atout pour l'édition littéraire ?

Dans le monde de l'édition littéraire française, rares sont les professionnels à connaître les programmes de la famille T_EX, plus rares encore ceux qui les utilisent. Faut-il s'en étonner, dans la mesure où ces programmes ont été conçus avant tout pour répondre aux exigences de l'édition scientifique ? Et d'aucuns sans doute d'estimer qu'ils ne sauraient répondre à celles de l'édition littéraire. À tort, car, s'il est un type d'ouvrage où ces logiciels permettent d'associer qualité typographique et gain de productivité, c'est bien celui des récits ou des essais. Au reste, quoi de plus naturel, pour composer de simples colonnes de texte, que le travail... en mode texte ?

D'un format texte à l'autre...

Bien sûr, les auteurs n'utilisent pas L^AT_EX, mais un logiciel de traitement de texte. Heureusement, il est simple de convertir au format L^AT_EX un fichier XML comme ceux produits par LibreOffice ou Microsoft Word : avec un programme en ligne de commande comme Writer2L^AT_EX, niveaux de titre, styles de caractère, notes et renvois sont exportés correctement sans requérir aucun travail. Dans la foulée, et avant toute intervention humaine, le fichier L^AT_EX obtenu peut faire l'objet d'un traitement automatisé qui va permettre de réduire considérablement le nombre d'interventions humaines à venir.

Mettons que dans l'urgence ou pour un client au budget limité, on doive se contenter de viser une qualité de composition équivalente à celle d'un livre de la collection « Blanche » de Gallimard d'environ 300 pages : sur la base d'une copie correctement stylée et d'une maquette déjà conçue, avec le script shell que j'ai écrit pour manipuler ladite copie et mes gabarits L^AT_EX, il ne me faut ainsi que quelques minutes à un peu moins d'une heure (avec beaucoup de malchance) pour obtenir le PDF destiné à l'impression.

La part de hasard, ici, tient surtout au nombre de pages creuses de moins de cinq lignes — un des rares accidents typographiques auquel Gallimard semble prêter attention, veuves, orphelines, divisions en bas de page étant par ailleurs légion dans les livres de l'éditeur. Je pense impossible, pour l'heure, d'automatiser le traitement de ce problème sans risquer de sacrifier le gris typographique ou la tenue du registre : le traitement des pages creuses trop courtes requiert donc un peu de travail.

Automatiser le traitement de certains accidents de composition

Il existe toutefois d'autres accidents potentiels de composition pour lesquels on peut envisager sans grand risque de mettre en place un traitement automatisé. Ainsi, il est possible d'interdire les lignes à voleur comme les fins de ligne ou de page désagréables telles « . À » ou « ? Se », grâce aux expressions rationnelles¹, ce qui permet de proposer une qualité déjà sensiblement supérieure à celle d'un « Blanche » au prix d'un modeste surcroît de travail². Bien sûr, les expressions rationnelles (ou « regex ») n'appartiennent pas en propre à L^AT_EX : les scripts d'InDesign permettent aussi d'y recourir. Il n'en reste pas moins que le WYSIWYG ne prédispose pas à la programmation³, alors qu'il s'agit du prolongement naturel du travail en mode texte.

Pourquoi choisir LuaT_EX

L^AT_EX n'est pas un programme mais un format — disons, pour simplifier, « un langage » —, qui suppose de faire appel à un moteur de composition : les principaux utilisés aujourd'hui sont pdfT_EX, X_ET_EX et LuaT_EX. Outre la gestion des polices variables, une raison de préférer LuaT_EX est l'incorporation d'un interpréteur Lua.

Parmi les multiples applications de ce langage de programmation, je mentionnerai la possibilité de procéder aux calculs liés à l'empagement dans le fichier source lui-même (le

1. Langage informatique concis servant à décrire des motifs de chaîne de caractères, afin, ici, d'apporter des modifications aux chaînes correspondantes dans le document traité. Par exemple, pour éviter qu'une ligne se termine par le premier mot d'une phrase de moins de cinq lettres, il suffit de rechercher le motif `([. ! ? : ...] \s[ : upper : ] \w{0, 3}) \s` pour le remplacer ensuite par `$1~`, ce qui substitue une insécable à l'espace suivant ce premier mot. On devrait toutefois améliorer le motif recherché pour tenir compte, notamment, de l'éventuelle présence de guillemets ou de parenthèses fermantes après le signe de ponctuation et de guillemets ou de parenthèses ouvrantes avant la grande capitale du premier mot, mais je ne voudrais pas rebuter le lecteur avec une expression rationnelle plus complexe que la précédente.

2. Pour un livre présentant de nombreux alinéas très courts, le traitement systématique visant à empêcher les lignes à voleur entraînera un nombre élevé de lignes lavées, surtout si la justification est étroite. Il vaut donc mieux n'appliquer ce traitement qu'à des alinéas *a priori* suffisamment longs (que bien sûr les regex permettent d'identifier). Même dans un contexte plus favorable, il faut s'attendre à quelques lignes lavées, qui pourront toutefois être repérées facilement sans devoir s'user les yeux, on le verra, ce qui permet de limiter généralement à moins d'une demi-heure le temps de travail supplémentaire.

3. Je n'ai pas connaissance de statistiques sur la question. Parmi mes confrères utilisant InDesign, toujours est-il que les plus nombreux avec qui j'ai pu m'entretenir utilisaient peu les scripts (voire pas du tout), que rares étaient ceux capables d'adapter un script existant à leurs besoins, et qu'aucun ne semblait capable d'en écrire un *ex nihilo*.

fichier texte servant à générer le PDF) : par la définition, en plus du format, de quelques variables (par exemple, « canon des ateliers », « imprimé luxe » ; ou « Honnecourt », « 9 divisions ») et le choix de la police de labeur ainsi que de la force de corps, un petit script interne en Lua permettra de calculer simplement non seulement les marges, mais également l'interlignage du texte courant en fonction de la hauteur du rectangle d'empagement, de la justification, du corps et de la hauteur d'x — ce qui rend inutile de les saisir manuellement ou de les importer depuis une feuille de calcul externe.

Mais le plus intéressant en matière de typographie reste sans doute la possibilité qu'offre LuaT_{EX} de détecter un grand nombre d'accidents de composition, ce dont se charge l'extension lua-typo de Daniel Flipo⁴, qui dans le PDF de sortie signalera en couleur veuves, orphelines, divisions en fin de page⁵, pages creuses de moins de cinq lignes, superpositions en début ou en fin de ligne de mêmes séquences de lettres, lignes à voleur, lignes creuses trop longues, pages commençant par le dernier mot court d'une phrase, lignes terminées par le premier mot court d'une phrase... Dès lors, lua-typo permet de gagner un temps considérable lors du travail de révision de la composition : un survol rapide du PDF permet de repérer les problèmes évoqués. Ne reste plus qu'à les traiter.

Bâcler un livre, quelle excuse ?

En somme, LuaL_AT_{EX}, laisse le choix aux éditeurs littéraires : continuer à se satisfaire — comme le font encore la plupart — de la médiocrité typographique actuelle, en économisant au moins des sommes non négligeables ; ou au contraire accepter de voir investir le temps généralement alloué à la mise en pages dans le soin apporté aux détails qui font la différence entre une composition moyenne et une composition de qualité. Concluons avec Olivier Bessard-Banquy : « [L]e discours convenu sur la démocratisation du livre comme synonyme d'inévitable relâchement esthétique est irrecevable⁶. »

4. À qui l'on doit encore notamment l'extension lettrine et le module french de l'extension babel, permettant de franciser la typographie des documents produits par L_AT_{EX}.

5. S'il est évidemment possible de traiter ces trois problèmes simplement par la variation automatique de l'interlignage, voilà qui bien sûr n'est pas digne d'un travail soigné. L'extension lua-widow-control s'efforce quant à elle d'éliminer veuves et orphelines en recomposant un ou plusieurs alinéas antérieurs, mais le résultat n'est guère probant pour des justifications moyennes : beaucoup trop de lignes lavées. Rien pour l'heure ne remplace ici l'être humain.

6. BESSARD-BANQUY, Olivier. « Le livre moche à la française ». *L'Esthétique du livre*, édité par Alain MILON et Marc PERELMAN, Presses universitaires de Paris Nanterre, 2010.