

Merci au Monsieur B.C. de m'accorder les 20 min suivants.

J'avais envie d'intituler mon exposé par cette interrogation : philologie numérique avec regret inévitables ou avec une méthodologie d'Open source ?

La liberté de modifier

Du point de vue du génie logiciel,

-----

Avant le Plan

Un mot rapide

Je vais vous présenter mon travail en cours sur la préface au Menteur avec le logiciel TXM. Une pratique extrême peut-être, mais elle permet de raffiner la transcription disons fac-similaire, et de répondre à une problématique :

De 1542 en 1550, comment Meigret fait progresser sons orthotypographie proposée ?  
Combien de variétés des typèmes les imprimeurs de Meigret engagent à cet effet ?

1.1

Vous connaissez déjà la requête par le langage CQL. Je montre ici seulement un ensemble des items illustratifs. Il y en a jusqu'ici trois types, @

## 1.2

On sait que dans la première moitié du 16<sup>e</sup> siècle, l'imprimerie commença à installer en France. Il n'est pas étonnant de voir qu'il y a deux K, et que deux E sont pris de différentes familles de police. Si je veille à cette précision à tel point, c'est que Meigret est intéressant dans la perspective de l'histoire du livre. C'était le cas quand Nina Catach associe ce grammairien avec le concept de l'orthotypographie. 50 ans après cette historienne de l'orthographe française, il faut rendre compte de cela avec la philologie numérique.

## 1.3

Montrer la feuille « Unicode »

Grâce au constat de B. Colombat, il est possible que la différence est à imputer à la police, italique ou romaine.

## 2.1

Les réactions du philologue J. Lallot m'inspirent à intituler cette partie : Questions de repassage(s)

Nous savons que c'est au besoin, mais qu'on a pas vraiment un dispositif efficace à cet effet.

Il me semble important de distinguer trois besoins ou concepts.

Je ne vais faire un discours sur leur distinction sémantique, mais voudrais montrer comment leur disjonction permet une philologie progressive ou d'Open source.

## 2.2

Je récupère le texte de Meigret sur le site Virga, et je garde volontiers la saisie plutôt régularisé de son auteur.

## 2.3

Nous pouvons désormais produire une édition avec une graphie voulue, avec celle de Vocabulaire qui aide à augmenter l'intelligibilité. Cette facilité est aussi laissée aux futurs éditeurs.

Au fond, le mode de gestion de versions change.

Sauf par une contrainte au niveau institutionnel, on n'est obligé d'établir une édition qui comporte toutes les informations. La réduction des infos au moyen d'un script XSLt est souvent nécessaire, mais ce script peut parfois être mal maintenu ou peu documenté lorsqu'il est offert par un autre. Pour moi en tout cas, la gestion des données, ou des graphies en l'occurrence, par le tableur est plus facile que d'en faire par la vaste technologie de XML. A cet égard, si l'on n'est plus contraint par un fichier XML unique, on regagne un plus grand contrôle sur l'édition et son rendu.

Il est à noter que la vaste technologie de XML peut contribuer à la difficulté de réutiliser le fichier source, puisque le langage XML accorde par nature une très grande créativité personnelle et puisse rendre moins ou peu lisible le fichier source.

### 3.2

Sa saisie régularisée vaut telle qu'elle car elle est suffisamment cohérente.

Nous savons que l'exigence de l'exactitude graphique varie d'un chercheur à l'autre. Par exemple, si je n'ai pas fixé l'objectif de tracer les typèmes ou caractères engagés par Meigret, la distinction entre <k> latin et <κ> grec est en soi frivole. Voyez surtout que je n'ai pas forcément raison, et que Bettens n'a pas forcément tort. Chacun a sa propre exigence, et c'est heureusement comme cela qu'un tel ou tel texte bénéficie de plusieurs intelligences investies. Simplement, il vaut mieux qu'on rend ces fichiers ouverts aux prochains perfectionnements des autres éditeurs.

Comme vous pouvez le voir dans cet exposé jusqu'ici, j'ai un profil un peu développeur. A ce titre, je propose de distinguer deux contextes où on dit l'exploitabilité.

L'exploitabilité au niveau de la requête textuelle, c'est la cohérence de la transcription. Si on laisse l'utilisateur se débrouiller avec la fonctionnalité minimale de Find du navigateur, cet usage trouverait incertaine la qualité de saisie. Pratiquement, il faut expliciter les choix d'Unicode. Par exemple, si je propose une édition où les a avec tilde, e avec tilde, o avec tilde ne sont pas respectés, je dois expliquer, par exemple, le constat que ces caractères ne substituent que les voyelles suivies des nasales. Comme, je l'ai évoqué, la séquence <é> composé de deux Unicode (U+0119 U+0301) ou composé de trois Unicode (U+0065 U+0328 U+0301) ne sont pas la même chose pour la machine.

L'exploitabilité augmente si on explicite ce genre de chose pour l'utilisateur.

L'exploitabilité au niveau de la réutilisabilité, (philologie numérique tout court,) je dirais que c'est rétrocompatibilité ("backward compatibility"). C'est comme cela que, les usagers ou éditeur prochains ont l'intérêt d'accumuler les efforts dans le même fichier source. (C'est au fonds ce que Serge Heiden appelle la philologie progressive.) Quand un fichier source n'est pas bien configuré, documenté et maintenu, les autres personnes auraient moins d'intérêt de commencer par là. Après tout, on peut toujours proposer une édition ultime, mais aussi, pourquoi pas, garder les investissements précédents, surtout quand on est obligé de faire les choix peu évidents. Voyez par exemple que je garde aussi les saisies de Bettens, à qui je dois la version de départ. Là, un maître d'ouvrage, un œil plus expérimenté, ou un éditeur futur peuvent intervenir, confirmer mon choix ou revenir à celui de Bettens. Comme les communautés du logiciel libre aiment le dire, il ne faut pas "reinventing the wheel", et il faut prévoir le perfectionnement du fichier source par les autres.

La philologie, ou le travail sur le langage en général, est en soi une activité sans fin a priori.

### 3.3

Passons au point 3.3 . Quand l'exploitabilité est garantie par la lemmatisation, l'exactitude ou l'harmonisation de graphies est une question d'un autre niveau, disons esthétique. Bref, il n'est plus nécessaire de sacrifier la transcription.

Bref, l'Open source n'est pas seulement fournir le fichier source. Écrire une documentation sur l'édition pour laquelle on a déjà beaucoup investi est un bon point de départ.