



**HAL**  
open science

## Un prototype de lecteur de vario-document

Yannis Haralambous, Johann Nonat

► **To cite this version:**

Yannis Haralambous, Johann Nonat. Un prototype de lecteur de vario-document. Documents Virtuels Personnalisables, ENSt Bretagne, Jul 2002, Brest, France. hal-02111976

**HAL Id: hal-02111976**

**<https://hal.science/hal-02111976>**

Submitted on 26 Apr 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Un prototype de lecteur de vario-document (Résumé de 4 pages)

Yannis Haralambous\*      Johann Nonat†

10 mai 2002

## 1 Introduction

On peut classifier les formats de fichier de document électronique suivant deux critères suivant contradictoires : la précision de description de la géographie de la page, et l'adaptativité au contexte de visualisation/impression. Ainsi l'on a les deux extrêmes : d'une part le format de fichier HTML qui est très adaptatif au contexte — en changeant, par exemple, la taille de la fenêtre du navigateur HTML, le texte et les éléments graphiques se mettent en page automatiquement, de même la plupart des navigateurs possèdent une fonctionnalité de choix de taille des caractères — et d'autre part le format PDF qui est extrêmement précis mais qui n'offre aucune possibilité d'adaptation de la géographie de la page au contexte, mis à part la mise à l'échelle globale du contenu de la page. Cette mise à l'échelle est peu pratique puisque lorsque les caractères deviennent facilement lisibles souvent la largeur du paragraphe dépasse la largeur de l'écran et la lecture du texte demande un défilement continu ; inversement lorsque la page tout entière est visible, le texte est trop petit pour être lu aisément à l'écran.

D'autre part aucun de ces deux formats ne permet une adaptativité plus « intelligente » au contexte : par exemple, sous certaines conditions de lecture on pourrait envisager une modification du contenu de la page. Une formule mathématique très large pourrait être segmentée, opération qui devrait rester sous le contrôle de l'auteur, seul responsable du contenu. De même, le rapport des tailles entre caractères pourrait varier sous certaines conditions : si dans un ouvrage normal les titres sont de corps 24 et le texte de corps 12, et donc le rapport des corps est de 2:1, il serait inutile de garder le même rapport lorsque le corps du texte aurait augmenté de six points, dans le cas, par exemple, d'une édition pour mal-voyants : nul besoin de composer les titres à 36 points, ils pourraient rester en corps 24.

De même, un phénomène très important en typographie traditionnelle est la « mise à l'échelle optique », c'est-à-dire la variation de la forme des caractères

---

\*Département Informatique, ENST de Bretagne

†D.E.A. CHMEST, ENST de Bretagne

tères suivant leur corps, par opposition à la mise à l'échelle photo-mécanique où les formes restent les mêmes à un facteur d'échelle près. Ainsi, dans le cas de l'écriture latine, on constate que les caractères deviennent plus larges et plus espacées, la différence entre traits épais et déliés moins importante, et que les ligatures se délient, lorsqu'on passe aux petits corps. Inversement, pour les grands corps, les caractères deviennent plus étroits, les traits épais suivent la mise à l'échelle alors que les traits déliés gardent pratiquement la même épaisseur, le crenage devient plus prononcé, etc. lorsqu'on passe aux grands corps.

On voit donc que l'adaptativité au contexte se situe à plusieurs niveaux : niveau macro-typographique, lorsqu'il s'agit de la mise en page globale du document qui varie suite à un changement du contexte ; niveau moyen, où un élément de taille moyenne (entre le mot et le paragraphe, par exemple une formule) doit être traité de manière spéciale ; niveau micro-typographique, lorsque la forme et le comportement des caractères change suivant le contexte.

On s'aperçoit que pour tenir compte de toutes ces variations on serait amené d'une part à surcharger d'informations le document et d'autre part on nécessiterait un navigateur/visualisateur extrêmement puissant, capable de mettre en page le document en agissant sur tous les niveaux, et ceci en temps réel.

Mais la difficulté ne s'arrête pas là. Éblouis par les promesses de marchands de logiciels «WYSIWYG» on oublie souvent qu'il y a des documents qui nécessitent des ressources informatiques très poussées pour être mis en page, puisant dans la linguistique, le traitement automatique des langues, etc. Pour ne citer qu'un exemple, la mise en page d'une *édition critique* avec plusieurs appareils critiques et appareils de notes, titres courants contenant diverses informations, notes et numérotations marginales, notes de fin de document, index et glossaires, nécessite des calculs importants faisant intervenir des composantes logicielles en faisant pas partie des navigateurs ordinaires. Une telle mise en page serait très éprouvante pour une machine de puissance moyenne, et l'effectuer en temps réel, en suivant les changements des tailles de la fenêtre du navigateur, relève de la science-fiction.

## 2 Les *vario-documents*

Pour résoudre les problèmes décrits à la section précédente, nous introduisons le concept de *vario-document*. L'idée est très simple : au lieu de demander au navigateur/visualisateur d'effectuer la mise en page à la volée, suivant le contexte de visualisation, on prépare en amont un grand nombre de mises en page différentes, classées suivant certains paramètres de contexte ; au moment de la lecture du *vario-document*, le logiciel choisit, suivant les valeurs de ces paramètres, la mise en page appropriée (ou une mise en page suffisamment proche) et utilise les fonctions classiques du format de fichier pour l'afficher.

Ce concept s'adapte à tout format de fichier existant, mais n'a un intérêt que si ce format de fichier est suffisamment précis : on peut ainsi parler de *vario-DVI*, *vario-PDF*, *vario-SVG*, etc. Le «*vario-module*» se placera entre le navigateur et le *vario-document*, recevra les valeurs des paramètres de contexte, choisira la

mise en page appropriée pour ces valeurs et fournira des données «statiques» au navigateur.

On peut imaginer plusieurs paramètres de contexte : les plus classiques seraient la largeur et hauteur de la fenêtre de visualisation, ainsi que le corps des caractères de «texte courant», c'est-à-dire le corps de base du document. Mais d'autres paramètres peuvent être utiles dans certaines situations : par exemple, le niveau de difficulté de lecture. En effet dans certaines langues divers mécanismes peuvent être mis en œuvre pour faciliter la lecture d'un texte suivant le niveau d'aptitude du lecteur. Ainsi, par exemple, en russe on peut afficher les accents toniques pour faciliter la lecture en haute voix du texte, en arabe et hébreu on peut afficher les voyelles courtes pour faciliter la compréhension et lecture du texte, en japonais on peut afficher les caractères des deux syllabaires (*katakana* et *hiragana*) pour indiquer la prononciation des caractères idéographiques, chose qui peut être faite de manière progressive, suivant l'ordre d'apprentissage de ces derniers dans le système scolaire japonais : ainsi, un paramètre de contexte tout à fait pertinent pour un texte japonais pourrait être l'âge (ou niveau linguistique dans le cas d'un lecteur non-Japonais) du lecteur.

### 3 La mise en œuvre

Pour mettre en œuvre la technologie des vario-documents, un certain nombre de problèmes doivent être résolus :

1. on nécessite une puissante méthode de compression et de stockage des données, adaptée aux variations de la mise en page en fonction de la variation des paramètres ;
2. les logiciels de production de fichiers DVI, PDF, SVG, etc. doivent être adaptés pour pouvoir produire massivement des mises en page à partir d'un document source, d'une feuille de style de mise en page générique et d'un certain nombre d'indications sur les variations possibles ;
3. les logiciels de visualisation de ces mêmes formats de fichiers doivent être adaptés pour communiquer avec le vario-moteur : ils doivent fournir à celui-ci les différents paramètres du contexte (soit obtenus directement à partir de la fenêtre de visualisation, soit par le biais d'éléments d'interface graphique, comme des boutons ou bandes de défilement), récupérer les données correspondantes et les afficher en temps réel ;
4. de plus, comme on peut, à juste titre, estimer que la majorité des auteurs de vario-documents ne voudront pas passer trop de temps sur la préparation de leur document, en ce qui concerne la prévention de problèmes de présentation dus à des valeurs de paramètres éloignées de la moyenne, un système de prévention devrait les alerter lorsqu'une situation exceptionnelle risque de se produire pour certaines de ces valeurs. Ainsi, l'auteur prépare son document de la manière habituelle, mais le système l'avertit que telle formule posera un problème lorsque la largeur de page sera inférieure à telle valeur, et lui propose de choisir parmi un certain nombre

d’alternatives, son choix devenant partie intégrante du vario-document. On nécessite donc des outils interactifs pour indiquer le comportement du vario-document sous l’effet de la variation des paramètres contextuels, et surtout dans le cas des valeurs extrêmes.

Notons qu’en ce qui concerne la compression des «vario-données», on peut atteindre des facteurs de compression importants à cause de certaines particularités de la composition typographique. En effet, alors que les formats de fichier DVI, PDF, SVG, etc. considèrent un paragraphe de texte comme un ensemble de caractères et de coordonnées, on constate que :

- les caractères forment des mots, et la plupart des variations contextuelles n’agissent pas à l’intérieur des mots : une variation de largeur de fenêtre de visualisation n’affectera en rien la position relative des lettres dans un mot, *sauf* si le mot est amené à être coupé. Mais l’opération de césure est statistiquement peu fréquente et peut être considérée comme le remplacement d’un mot par deux sous-mots, auxquels le même principe de non-variation relative s’applique. Ainsi, on peut considérer, à quelques exceptions près, qu’un la variation d’un paragraphe peut être décrite uniquement par la variation des position des débuts de mots, ce qui réduit déjà la quantité d’informations nécessaires ;
- autre facteur, encore plus important : une grande partie de la variation d’un paragraphe selon des paramètres à variation continue, peut être *interpolée* : en effet, lorsqu’un on change de  $\varepsilon$  la justification d’un paragraphe contenant  $n$  mots, et donc  $n - 1$  blancs, la largeur des blancs va changer de  $\frac{\varepsilon}{n-1}$ , *sauf* si cette variation de justification entraîne une césure ou un passage d’un mot à la ligne, dans lequel cas la géographie du paragraphe peut changer complètement. Entre les différentes répartitions des mots (ou de sous-mots, sous l’effet d’une césure) sur les lignes, la variation peut donc être décrite par une interpolation, ce qui réduit énormément la quantité d’informations nécessaires ;
- troisième facteur, la *mésure d’acceptabilité typographique*. En effet, pour obtenir des mises en pages acceptables typographiquement, on peut poser certaines contraintes, qui limitent encore le nombre de versions possibles d’un même document. Des logiciels comme  $\text{\TeX}$  (et son extension,  $\Omega$ ) permettent de mesurer la qualité typographique de chaque paragraphe ; on peut alors demander que toutes les mises en page contenues dans un vario-document soient de «qualité typographique» supérieure à une certaine valeur. Cela évite les mises en page «maladroites» et réduit davantage le nombre de versions contenues dans un vario-document.

## 4 Un prototype le lecteur de vario-DVI

DVI (acronyme de *Device Independent*) est le format de fichier de sortie de  $\text{\TeX}$ . Il date des années ’70 et a été conçu comme une abstraction de la géographie de la mise en page : dans un fichier DVI seuls sont indiqués les positions des

caractères et les coordonnées de rectangles noirs (pour produire des filets, etc.). Les glyphes (en contours ou en bitmap) et les figures n'entrent en jeu que lorsque le format DVI est converti en un autre format, comme par exemple le langage PostScript.

La simplicité du format de fichier DVI en fait le candidat idéal pour le prototypage des vario-documents. D'autant plus que le logiciel idéal de production de vario-documents est  $\Omega$ , co-développé par le premier auteur.

Il existe un logiciel libre de lecture de fichiers DVI sous le système d'exploitation Unix avec système de fenêtrage X Window : il s'agit de `xdvi`. Ce logiciel accède directement aux glyphes et aux figures PostScript et les affiche dans une fenêtre X Window.

Nous avons modifié `xdvi` pour le rendre compatible avec les vario-documents. Dans notre prototype nous avons ajouté à l'interface graphique de `xdvi` des boutons pour modifier les valeurs de trois paramètres : la largeur et la hauteur de la fenêtre, ainsi que la taille standard des caractères.

Ce prototype démontre l'utilité du concept de vario-document et permet de mesurer l'efficacité des diverses méthodes expérimentales de compression et de stockage de vario-données, mises en œuvre dans le cadre du projet général de développement de vario-outils.

[Note au referee : dans la version complète de l'article nous ajouterons également un grand nombre de figures illustrant les différents concepts décrits ou mentionnés dans l'article, ainsi que des copies d'écran du prototype `xdvi`.]