

# *Cahiers* **GUT** *enberg*

## ∞ L'ÉDITION STRUCTURÉE ET LE WORLD-WIDE WEB

¶ Vincent QUINT, Irène VATTON

*Cahiers GUTenberg*, n° 19 (1995), p. 85-97.

<[http://cahiers.gutenberg.eu.org/fitem?id=CG\\_1995\\_\\_19\\_85\\_0](http://cahiers.gutenberg.eu.org/fitem?id=CG_1995__19_85_0)>

© Association GUTenberg, 1995, tous droits réservés.

L'accès aux articles des *Cahiers GUTenberg*

(<http://cahiers.gutenberg.eu.org/>),

implique l'accord avec les conditions générales

d'utilisation (<http://cahiers.gutenberg.eu.org/legal.html>).

Toute utilisation commerciale ou impression systématique

est constitutive d'une infraction pénale. Toute copie ou impression

de ce fichier doit contenir la présente mention de copyright.



# L'édition structurée et le World-Wide Web

---

Vincent QUINT (INRIA)

Irène VATTON (CNRS)

*INRIA-IMAG, 2 avenue de Vignate, 38610 Gières, France*

**Résumé.** La création de documents pour le World-Wide Web n'est pas toujours une tâche facile. Beaucoup d'auteurs créent ces documents « à la main ». Ils doivent alors saisir la syntaxe HTML, même si l'éditeur de texte qu'ils utilisent fournit quelques aides. Une alternative consiste à utiliser les filtres de différents systèmes de production de documents, mais ces systèmes n'intègrent pas toutes les spécificités du Web. Ces deux méthodes ne sont donc pas complètement satisfaisantes.

Nous présentons ici une solution fondée sur l'éditeur de documents structurés Grif. L'éditeur Grif a été étendu pour prendre en compte les caractéristiques propres du Web et en faire un environnement confortable pour la création des documents sur le Web.

## 1. Introduction

Le World-Wide Web (www) est un service hypermédia réparti, accessible par le réseau Internet. Il permet à tout utilisateur de ce réseau d'accéder à travers un logiciel client à un très grand nombre de documents électroniques qui sont gérés par des serveurs répartis sur toute la planète. Les échanges entre serveurs et clients sont fondés sur une représentation commune des documents, HTML (HyperText Markup Language).

Depuis quelque temps, ce service connaît un succès foudroyant, avec des taux de croissance exponentiels (trafic, nombre d'utilisateurs, nombre de serveurs, etc.). Une grande part de ce succès vient de la simplicité des concepts et protocoles mis en jeu, ce qui a permis un développement relativement rapide de logiciels serveurs et clients de qualité. Les clients, pour l'essentiel des « browsers » comme Mosaic, Netscape ou Arena, ont fortement contribué à ce succès, en offrant une interface utilisateur simple d'emploi et en intégrant un choix intéressant de services.

Mais tous les documents auxquels accèdent des millions d'utilisateurs doivent être écrits par des auteurs, même si quelques uns sont produits automatiquement par des programmes. Et sur ce point, la situation est moins réjouissante. Les outils utilisés couramment manquent de puissance, de souplesse et de simplicité d'emploi. C'est ce problème que nous essayons de traiter dans cet article. Notre objectif est d'offrir aux auteurs le même confort et les mêmes avantages que ceux dont profitent déjà les lecteurs.

Dans ce but, nous proposons d'utiliser un éditeur structuré, Grif, pour construire un environnement d'auteur spécifiquement destiné à WWW. Nous analysons d'abord dans la section 2 les caractéristiques propres des documents HTML. Nous faisons ensuite, dans la section 3, une analyse critique des outils proposés pour la production de ces documents. La section 4 présente l'environnement que nous avons construit à partir de Grif. Enfin la section 5 montre les avantages que cet environnement peut apporter non seulement aux auteurs, mais aussi aux lecteurs.

## 2. Les documents actuels

HTML est un langage de marquage conçu pour représenter la structure des documents, mais il est souvent utilisé comme un langage de présentation. Comme il doit être capable de s'adapter à une grande variété de documents, il définit des types d'éléments très généraux. La structure qu'il propose est donc très simple et imprécise. Par exemple, il propose des titres de sections (éléments H1, H2, ..., H6) pour exprimer des niveaux de titres différents, mais il n'impose aucune règle sur l'ordre d'utilisation de ces différents titres. Rien n'empêche par exemple une séquence de titres H1, H6, H3. Pratiquement, la seule hiérarchie imposée est celle des éléments de liste (LI) dans les différents types de listes (UL, OL, etc.).

Cette structure très lâche présente en fait un avantage pour beaucoup de documents du Web. Ces documents sont souvent courts et contiennent beaucoup de liens vers d'autres documents. Mais il est aussi utile de rendre accessible sur le Web de gros documents, par exemple des documents techniques (manuels, articles scientifiques, etc.). Aujourd'hui ces documents sont disponibles sous deux formes.

1. Comme un ensemble de pages HTML. Cette forme est bien adaptée à la lecture sur écran, mais rend difficile le traitement des documents hors du Web. De plus, le document complet a perdu une partie de sa structure.
2. Dans un format non HTML (PostScript, dvi, etc.). Les documents n'étant pas au format HTML, ils ne peuvent pas être lus avec les outils de base du Web, mais ils gardent leur intégrité.

Il est important de maintenir sur le Web les deux types de documents : d'une part des documents courts pour gérer l'accès à l'information pertinente et d'autre part des documents techniques bien structurés pour fournir des informations plus complexes. Même si HTML permet de représenter les deux catégories de documents, les outils actuels du Web ne sont pas vraiment adaptés aux documents complexes.

### 2.1. Des documents mal structurés

Contrairement à ce que peuvent laisser croire beaucoup de documents présents sur le Web, HTML n'est pas simplement un jeu de marques insérées au fil du texte.

Il y a des règles clairement énoncées dans une DTD SGML (Document Type Definition) et dans les documents de référence qui l'accompagnent [2]. Malheureusement, beaucoup de documents sont produits « à la main »<sup>1</sup> sans que l'auteur se réfère à la DTD. La vérification de la conformité des documents se limite bien souvent au simple test de leur apparence sur un ou deux outils de visualisation.

Ce n'est pas que la plupart des documents soient non conformes à la DTD HTML qui pose problème, mais le fait que la structure de ces documents peut devenir ambiguë. En effet les « browsers » que l'on utilise généralement pour lire les documents HTML sont très permissifs et ne rejettent que très rarement les documents mal structurés. Par contre, l'interprétation de la structure du document peut varier d'un outil à l'autre et donc le respect de l'intention de l'auteur n'est pas garanti.

## **2.2. Des liens mal gérés**

Les liens constituent un élément essentiel dans les documents du World-Wide Web. Malheureusement, aujourd'hui aucun outil n'offre une aide réelle pour la création et la mise à jour de ces liens. L'absence d'outils qui exploitent toute la puissance de HTML sur les liens explique en partie l'existence de documents comportant des liens erronés ou mal typés.

Différents types de liens sont définis dans HTML, pour établir et typer les relations qui lient les documents :

- des liens hypertextuels libres, qui sont les plus fréquents ;
- mais aussi des liens structuraux, pour exprimer que tel document représente une description détaillée ou bien la suite d'un autre document, etc.

Les liens structuraux sont très utiles pour matérialiser l'organisation d'un gros document éclaté en petits documents HTML et leur sémantique est clairement définie. La méconnaissance de la DTD HTML et l'absence d'outils de création spécifiques expliquent que ces liens ne soient pratiquement jamais exploités. On trouve même des documents qui utilisent des liens hypertextuels libres pour matérialiser des relations structurales entre documents, où la sémantique du lien est exprimée dans le texte ou par l'icône associée au lien.

L<sup>A</sup>T<sub>E</sub>X, FrameMaker, Word, etc. proposent des filtres pour traduire en HTML les documents qu'ils produisent. Ces filtres sont très utiles pour mettre sur le Web la documentation existant par ailleurs, mais ils ne permettent pas de faire du système de production de documents un environnement complet de création de documents HTML. Dans la mesure où ces systèmes ignorent l'existence du Web, il ne peuvent pas prendre en considération toutes ses spécificités, comme par exemple l'interconnexion entre documents sur le réseau.

---

1. Les éditeurs de texte comme Emacs offrent des macros qui simplifient la saisie de la syntaxe HTML, mais n'aident pas réellement l'auteur à construire des documents HTML corrects.

### 3. L'édition SGML et le World-Wide Web

Construire un environnement auteur pour le World-Wide Web à partir de rien représente un travail considérable. Comme HTML est basé sur la norme SGML, il semble raisonnable d'utiliser des outils SGML pour construire un tel environnement. Une solution rapide et naïve consiste à prendre un éditeur SGML paramétré par la DTD HTML. C'est typiquement la solution adoptée par HoTMetaL. Cette solution fonctionne, évidemment, mais elle n'est pas satisfaisante parce que non adaptée au contexte particulier du Web, qui présente les caractéristiques suivantes.

- Les documents présents sur le Web sont rarement conformes à la DTD HTML<sup>2</sup>. Ces documents sont donc rejetés par tout éditeur SGML standard. L'outil mis en œuvre ne peut pas se contenter de créer de nouveaux documents. Il doit être capable de charger, voire même de modifier, des documents existants mal structurés.
- HTML est très peu structuré. Sauf dans les listes, il est possible d'insérer à peu près n'importe quel élément n'importe où dans le document. Un éditeur générique SGML, qui construit son interface utilisateur sur la simple base des DTDs qu'il manipule, sans aucune autre adaptation, produit un dialogue peu adapté à un utilisateur novice dans le monde de l'édition structurée.

Un moyen de minimiser le nombre de documents rejetés (voir le point précédent) consiste à assouplir la DTD ; ce qui a tendance à aggraver le problème de l'interface utilisateur.

- Les documents HTML contiennent beaucoup d'ancres qui sont des composants spécifiques au Web. Ce sont les points de départ et d'arrivée des liens hypertexte. Un environnement auteur pour le Web se doit de fournir une aide à la création et la mise à jour de ces ancres. Les outils SGML classiques ne fournissent rien de ce genre.
- Les auteurs de documents HTML attachent beaucoup d'importance à l'apparence de leurs documents. Il faut donc que l'outil ait les mêmes capacités de visualisation que les « browsers » utilisés pour la simple consultation. Avec les extensions dans les domaines des tableaux et des mathématiques proposées pour la version 3.0 de HTML, ce problème va devenir encore plus crucial.

Il n'existe pas aujourd'hui d'environnement auteur satisfaisant pour le Web. Les éditeurs de texte sont certes faciles à utiliser, mais ils n'apportent aucune aide réelle à l'utilisateur. Celui-ci doit être un expert HTML, surtout s'il s'agit de produire des formulaires, des tableaux ou d'autres objets complexes. Les systèmes de production de documents, accompagnés de leurs filtres, rendent beaucoup de

---

2. La DTD actuellement en vigueur est HTML 2.0. Des versions précédentes ont été utilisées, mais bien des documents ne sont conformes à aucune de ces DTD.

services, mais leur mise en œuvre est lourde et leur rôle reste limité. L'édition SGML peut répondre au besoin à condition de s'adapter aux conditions particulières du World-Wide Web.

## 4. Un environnement auteur fondé sur Grif

Grif [3] est essentiellement un éditeur de documents structurés dont les principes sont inspirés de l'édition syntaxique. Un modèle (une DTD) spécifie la structure du type de document traité et l'éditeur utilise ce modèle pour aider (et parfois obliger) l'utilisateur à produire un document conforme au modèle. Un modèle permet de décrire l'organisation hiérarchique du document ; par exemple, il indique qu'un document de type article est composé de sections, sous-sections et paragraphes. Un modèle permet aussi de définir des relations non hiérarchiques [4] à l'intérieur d'un même document ou entre documents. Ce sont typiquement des liens comparables à ceux de HTML. Grif intègre en fait tous les mécanismes de base nécessaires à un bon éditeur de documents SGML et il permet de construire un environnement auteur complet pour le World-Wide Web.

### 4.1. Environnement de base

L'interface utilisateur de Grif est fondée sur le concept de manipulation directe. L'utilisateur peut donc éditer les documents sous leur aspect formaté plutôt que de manipuler la syntaxe SGML ou HTML. En effet, connaissant la structure logique des documents manipulés, il est possible de définir de façon générique l'aspect graphique des documents construits sur ces modèles. Grif associe à chaque modèle de structure un modèle de présentation qui indique comment présenter chaque type d'élément en fonction de sa position dans la structure : corps et style des caractères, placement, choix des couleurs, etc. Plusieurs modèles de présentation peuvent être associés au même modèle de structure, permettant ainsi de changer globalement la présentation d'un document, simplement en changeant de modèle de présentation. Cette fonction est très utile pour adapter la présentation des documents HTML lorsqu'ils sont utilisés comme support d'un exposé (plus de couleur, des corps de caractères plus gros, etc). Elle permet aussi d'adapter la présentation des documents lors de leur impression (mise en page, corps de caractères plus faible).

Dans chaque modèle de présentation, il est possible de définir plusieurs vues du document. Chaque vue présente dans une fenêtre distincte la même structure logique, mais avec des points de vue différents. Ainsi, pour les documents HTML, il est possible de présenter en même temps une vue formatée du document et la vue du source HTML, comme le montre la figure 1.

Grif ne se restreint pas aux documents structurés textuels. Il sait de la même façon afficher et manipuler des structures complexes comme les tableaux et les formules mathématiques. Il sait aussi manipuler des documents contenant des

images de différents formats (XBM, XPM, Gif, PostScript, etc.). C'est donc un environnement complet pour créer, manipuler et imprimer des documents structurés complexes.

Un autre intérêt de Grif réside dans sa capacité d'ouverture et d'extension. Une API permet à tout programme d'accéder aux fonctions d'édition qui sont proposées aux utilisateurs humains. Il est donc possible d'étendre les fonctions natives de l'éditeur : une application peut indiquer quels sont les événements qui l'intéressent pour que l'éditeur lui donne le contrôle quand ces événements se produisent. Des traitements spécifiques peuvent ainsi être ajoutés à l'éditeur [5].

## **4.2. Adaptation de Grif**

Tel quel, Grif présente un environnement auteur confortable pour le Web. Il sait traiter toutes les structures de HTML et permet de travailler sur des documents formatés. Mais les spécificités des documents HTML causent certains désagréments : beaucoup de documents disponibles ne peuvent pas être chargés, les liens ne sont pas traités en tant que tels, etc. De plus l'interface utilisateur de Grif, conçue pour des documents fortement structurés, est un peu lourde. Pour toutes ces raisons, des améliorations spécifiques ont été apportées à l'éditeur.

### *4.2.1. Interface utilisateur*

Comme la DTD HTML est très plate, les menus que Grif construit pour offrir le choix des éléments à créer sont le plus souvent constitués d'une longue liste qui offre presque tous les types d'éléments disponibles dans HTML. Pour présenter des menus plus utilisables, deux techniques complémentaires ont été mises en œuvre. La première consiste à modifier explicitement l'interface utilisateur de l'éditeur, la seconde à aménager le modèle de structure de HTML.

Les mécanismes d'extension de l'éditeur ont été utilisés pour modifier l'interface utilisateur. Ils permettent en effet à des programmes d'application d'adapter l'éditeur à leurs propres besoins, et donc de modifier l'interface utilisateur standard (barre de menus, menus déroulants, etc.).

La deuxième technique a consisté à transposer la DTD HTML dans les termes des structures manipulées par Grif. En particulier, les éléments que la DTD originale place sur le même plan ont été regroupés en catégories (titres, listes, paragraphes, etc.). Ainsi, au lieu de choisir dans un seul long menu, l'utilisateur détermine d'abord la catégorie dans un court menu, puis il choisit, dans un deuxième menu, le type exact dans cette catégorie. L'utilisation de ces deux niveaux de menus est beaucoup plus efficace.

D'autres adaptations de la structure HTML ont été faites pour faciliter la tâche de l'auteur. Les éléments comme B (gras), I (italique), EM (emphase) ont été remplacés par des attributs. L'interface utilisateur de Grif se prête beaucoup mieux



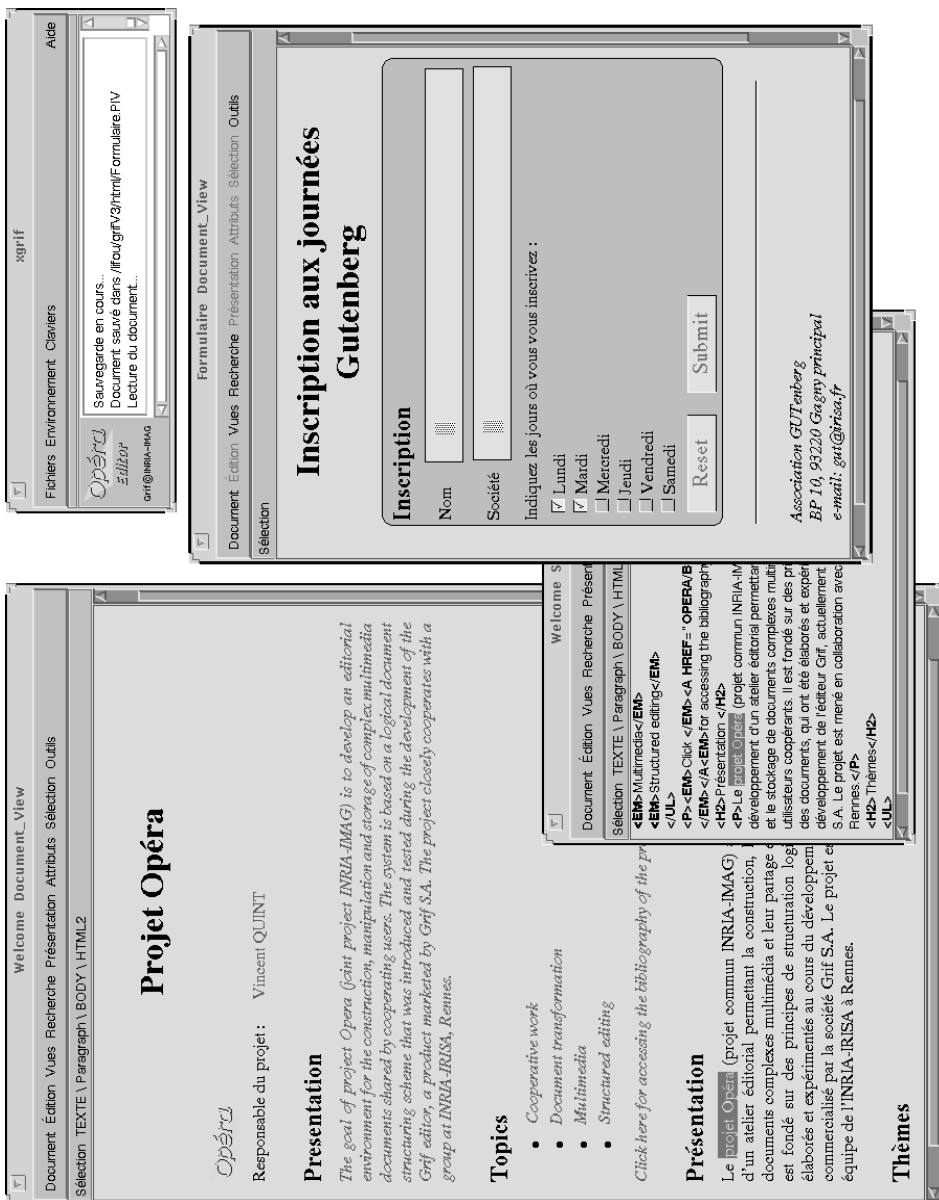


FIGURE 1 - Deux documents HTML en cours d'édition

à la manipulation sous forme d'attributs de ce type d'information qu'on ajoute et qu'on retire souvent sur du texte existant.

La création de formulaires par des auteurs de documents HTML a également été simplifiée. L'unique type INPUT, qui représente aussi bien différents types de boutons (basculer, radio, soumission, remise à zéro, etc.) que différentes zones de saisie (texte, image, mot de passe) a été remplacé par des types d'éléments différents pour Grif, rendant les choses plus claires à l'utilisateur.

Avec ces adaptations du modèle de structure, les traitements internes de l'éditeur ne se font pas exactement de la façon spécifiée par la DTD HTML, mais, sous leur forme externe, les documents sont strictement conformes à cette DTD.

#### *4.2.2. Saisie des liens*

Dans HTML, un lien est une relation entre une ancre de départ et une ancre d'arrivée, chacune englobant une partie de document, le plus souvent une simple chaîne de caractères. La relation proprement dite est indiquée par des attributs : un attribut NAME est associé à l'ancre d'arrivée et identifie cette ancre de façon unique dans le document ; un attribut HREF est associé à l'ancre de départ et indique l'identificateur (attribut NAME) de l'ancre d'arrivée du lien. Dans le cas d'un lien d'un document vers un autre document, l'attribut HREF contient l'adresse réseau (URL) du document d'arrivée et éventuellement de l'ancre d'arrivée (dans le cas où c'est un point du document qui est la cible du lien, et non le document dans son ensemble).

Il est possible de laisser l'utilisateur manipuler les liens dans Grif en suivant ce modèle, tel qu'il apparaît dans la DTD. C'est d'ailleurs ce que fait HoTMetaL : l'utilisateur frappe au clavier les valeurs des attributs NAME et HREF, avec tous les risques d'erreur que cela comporte. Il se trouve que Grif a également une notion de lien hypertexte et qu'il traite aussi bien les liens internes (départ et arrivée dans le même document) que les liens externes. L'établissement du lien entre deux éléments se fait simplement en cliquant les éléments concernés.

Cette interface est également proposée, et les mécanismes d'extension de l'éditeur ont été mis à profit pour affecter automatiquement les bonnes valeurs aux attributs. L'attribut NAME est ainsi engendré par l'éditeur et son unicité est garantie. L'attribut HREF est également mis par l'éditeur qui utilise soit l'URL du document cible (lien vers un autre document) soit l'attribut NAME de l'ancre d'arrivée.

Cette façon d'établir les liens externes est rendue possible par l'utilisation de la bibliothèque www dans l'éditeur Grif. L'éditeur peut ainsi charger des documents distants, comme le fait un « browser », et l'utilisateur peut dès lors les manipuler localement, et en particulier les cliquer pour établir des liens simplement et sans risque d'erreur. Ces documents peuvent aussi être utilisés par la commande copier, pour insérer certaines de leurs parties (y compris des liens) dans les documents en cours de rédaction.

### 4.2.3. Structure des documents

Pour permettre aux auteurs de travailler sur des documents existants, l'éditeur doit accepter tous les documents, même ceux qui ne sont pas strictement conformes à la DTD HTML. Mais, comme tout éditeur SGML, Grif vérifie les documents qu'il lit et n'accepte que ceux qui satisfont les contraintes exprimées dans la DTD. Or WWW n'est pas un environnement SGML typique. Au contraire, il demande beaucoup de souplesse et de tolérance vis à vis de la structure des documents. C'est pourquoi nous avons relâché les contrôles effectués par l'éditeur lors de la lecture des documents.

L'éditeur doit en conséquence travailler sur des documents qui ne sont pas conformes à leur modèle de structure, mais il peut le faire sans difficultés. Évidemment, lorsqu'il crée de nouveaux éléments, ces éléments ne sont insérés que là où le modèle de structure l'autorise, mais les éléments existants ne perturbent pas l'éditeur s'ils sont mal placés. Ce comportement fait que Grif peut accepter les documents HTML comme ils sont ; il garantit cependant que ces documents ne seront pas dégradés par de nouveaux traitements.

L'édition structurée des documents de WWW ne rend pas seulement l'écriture plus confortable, elle permet aussi une lecture plus efficace. Grif est utilisé pour gérer l'environnement de l'utilisateur avec toute la puissance offerte par la structuration. Les listes de documents intéressants (*hot lists*) peuvent être organisées de façon hiérarchique, pour aider l'utilisateur à trouver très rapidement les documents qu'il cherche. Chaque annotation peut être structurée et peut être représentée et traitée comme un document HTML, ce qui permet d'organiser les commentaires et d'y inclure des liens vers d'autres documents. Comme elles sont traitées par l'éditeur, ces listes et annotations peuvent être consultées, réarrangées ou modifiées par l'utilisateur à tout moment, ce qui constitue un avantage appréciable.

### 4.3. Transformations de structure

Comme on l'a vu, Grif accepte les documents mal structurés, mais il permet aussi à l'utilisateur de restructurer ces documents à l'aide de différentes commandes.

Les commandes habituelles d'édition permettent naturellement de manipuler la structure et donc de corriger une structure erronée. Un élément qui n'est pas à la bonne position structurale peut être déplacé par les commandes Couper et Coller, la commande Coller vérifiant que la structure qu'elle insère est bien licite à l'endroit considéré. Un élément obligatoire mais absent peut être ajouté par la commande Insérer, un attribut invalide peut être retiré ou modifié, etc.

D'autres commandes sont spécifiquement destinées à transformer une structure existante : Diviser, Changer, Englober. Comme toutes les commandes offertes par l'éditeur, elles garantissent que la structure qu'elles produisent est conforme au modèle de structure.

La commande *Diviser* permet d'éclater un élément d'un type donné en deux éléments de ce type, une partie du contenu de l'élément d'origine passant dans le deuxième élément (l'autre partie reste dans le premier élément). Dans HTML, cela permet par exemple de couper une liste en deux.

Les commandes *Changer* et *Englober* sont davantage utilisées pour corriger des erreurs de structure. La première permet de changer le type d'un élément présent dans un document, sans le déplacer. Elle présente à l'utilisateur la liste des types dans lesquels l'élément sélectionné peut être changé. Cette liste contient tous les types définis dans le modèle de structure du document, avec lesquels le type de l'élément est isomorphe et qui peuvent apparaître à la position structurale où se trouve l'élément dans le document. Dans HTML, cela permet, par exemple, de transformer une liste non numérotée (UL) en une liste numérotée (OL), ou un titre d'un certain niveau (H*n*) en un titre d'un autre niveau. En fait, la DTD HTML, comme elle est très plate, permet un grand nombre de telles transformations.

La commande *Englober* insère un nouvel élément dans l'arbre d'un document, en gardant les éléments qui se trouvaient à cette position de l'arbre et en en faisant les descendants du nouvel élément. Dans un document HTML, cette commande sert par exemple à créer un élément *Paragraphe* (P) qui englobe une chaîne de caractères, ou à créer une liste (UL) à partir d'une séquence de paragraphes.

La commande *Coller* est aussi une commande de restructuration. Quand un utilisateur veut coller une partie d'un document dans ce même document ou dans un autre, l'opération peut poser un problème : la structure de la partie à coller peut violer les contraintes que le modèle de structure impose à la position où on veut coller. La partie à coller doit donc être restructurée. Le problème dans sa généralité est complexe [1], mais, s'il s'agit de traiter des documents HTML, il peut être considérablement simplifié, par une approche fondée sur des tables de conversion. Ces tables indiquent comment chaque type d'élément doit être converti. Dans le cas de HTML, elles peuvent être construites « à la main » : la DTD est petite et les structures qu'elle définit sont simples. La construction automatique de ces tables est à l'étude, mais il s'agit là d'un travail d'une autre ampleur.

## 5. Utilisation de l'environnement auteur

L'environnement d'édition constitué par Grif et les adaptations indiquées section 4 offre un certain nombre de possibilités nouvelles aux auteurs de documents HTML.

### 5.1. Traitement des gros documents

La plupart des filtres permettent de diviser un gros document en un ensemble de petits documents HTML reliés entre eux par des liens hypertexte. Cette forme

convient bien à la lecture interactive à l'aide de « browsers », mais la lecture sur papier s'en accommode mal.

Pourtant, HTML offre les moyens d'exploiter plus efficacement ces documents éclatés. En particulier, l'élément LINK, avec les attributs REL et REV, est fait pour décrire une structure arborescente qui relie les documents élémentaires, permettant ainsi de décrire l'organisation du document original. C'est ce que fait Grif, lorsqu'il transforme en HTML un document (voir section 5.2), conforme à un modèle de document plus complexe que celui qui a été spécifiquement conçu pour HTML.

Lorsqu'un document HTML éclaté est imprimé, Grif peut alors utiliser ces liens pour reconstruire la structure du document complet et l'imprimer dans sa totalité. De plus, en choisissant un modèle de présentation adapté, le document peut être imprimé avec un aspect graphique qui convient mieux au support papier (caractères plus petits, pages numérotées, titres courants, notes de bas de page, etc.).

Beaucoup de gros documents contiennent des équations ou des tableaux. La future version de HTML prévoit ce genre d'éléments, qui sont représentés selon une structure plus complexe que le reste des documents HTML. Comme Grif manipule déjà, de façon native, ces éléments, l'adoption de cette évolution de HTML ne pose pas de problèmes. L'édition structurée présentera même un avantage important dans ce cas, puisqu'elle est précisément destinée à simplifier la manipulation de structures complexes.

## **5.2. Conception de documents**

L'approche structurée apporte un certain nombre d'avantages dans la conception des documents. Grâce à la notion de modèle de structure, on peut garantir que des documents de même type auront non seulement un aspect graphique homogène, mais aussi une structuration semblable.

On peut ainsi définir des modèles de structure et de présentation pour les différents types de documents que l'on veut produire pour WWW. Dans un centre de recherche, par exemple, on définira des documents techniques, des rapports de recherche, des documents de présentation des équipes, des projets, des personnes, etc.

La traduction de ces documents en HTML se fait avec le mécanisme d'exportation de Grif. Celui-ci permet en effet de produire une sortie des documents qui soit conforme à une syntaxe spécifiée dans un modèle de traduction, à l'aide d'un langage appelé T. C'est ce qui a été utilisé, par exemple, pour produire les versions L<sup>A</sup>T<sub>E</sub>X et HTML de cet article. Pour chaque modèle de document, on spécifie donc un modèle de traduction vers HTML. Cela permet alors de sortir en HTML des documents qui répondent à des modèles de structure différents et, pour chaque modèle de structure, on obtient des formes HTML homogènes, puisque construites à partir de la même structure logique, par application des mêmes règles de traduction.

Cette façon de produire du HTML peut aussi être utilisée pour de gros documents structurés. Un modèle de structure peut être défini pour ces documents et, lorsqu'ils sont exportés en HTML par Grif, ils peuvent être découpés en plusieurs petits documents, liés ensemble (voir section 5.1). Ce service offert par Grif peut être comparé à celui d'un filtre. La différence importante est que le document HTML produit peut être immédiatement chargé dans le même environnement et à nouveau manipulé, par exemple pour y ajouter de nouveaux liens vers des documents résidant sur des serveurs éloignés (voir section 4.2.2).

### **5.3. Transition vers SGML**

Grif peut traiter simultanément plusieurs documents, selon plusieurs modèles de structure. Même avec les adaptations propres à WWW (voir section 4.2), il n'est pas restreint aux seuls documents HTML. Ainsi, les utilisateurs peuvent traiter dans le même environnement des documents de différents types et des documents HTML.

On l'a déjà vu, des documents construits selon d'autres modèles de structure peuvent être sortis en HTML, mais on peut aussi imaginer que des documents SGML soient échangés directement sur le Web, sans passer par le format HTML. Cela permettrait notamment de tirer parti de toutes les possibilités de structuration offertes par SGML, pour tous les utilisateurs disposant d'outils SGML. Les documents SGML pourraient alors être utilisés à l'intérieur de petites communautés d'utilisateurs qui travaillent sur des documents particuliers, les documents HTML étant utilisés par l'ensemble de la population qui accède au Web. Certains documents pourraient même être publiés sous les deux formes.

De cette façon, on peut envisager une évolution progressive de WWW vers SGML, comme le propose [6].

## **6. Conclusion**

L'environnement auteur présenté dans cet article permet un nouveau mode de production des documents destinés au World-Wide Web. Les auteurs travaillent sur une représentation formatée et très lisible des documents, même pour les parties les plus complexes, comme les formulaires et bientôt (dès que HTML 3.0 sera adopté) les tableaux et les formules mathématiques. Ils n'ont pas à se préoccuper de la syntaxe HTML, ni à vérifier la structure de leurs documents. Tout cela est pris en charge par l'outil. Ils peuvent aussi manipuler confortablement les liens entre documents répartis.

Cet environnement maintient une complète compatibilité avec les autres outils, puisqu'il peut lire tous les documents déjà disponibles sur le Web, même ceux qui ne sont pas strictement conformes à la DTD HTML. Il garantit aussi que tous les documents qu'il produit (ou les parties qu'il ajoute dans les documents existants) sont corrects vis à vis de cette DTD.

L'approche structurée, même pour des documents peu structurés comme ceux de WWW, apporte des avantages certains et permet une évolution sans rupture vers une forme plus riche des documents échangés. Elle ouvre les portes du Web aux documents SGML. Elle permet aussi à des documents simples de coexister avec des documents plus complexes. Les choses simples restent simples, les choses complexes deviennent plus simples.

## Références bibliographiques

- [1] E. Akpotsui, V. Quint, « Type Transformation in Structured Editing Systems », *Proceedings of Electronic Publishing 1992, EP92* (C. Vanoirbeek et G. Coray, ed.), 27–41, Cambridge University Press, April 1992.
- [2] T. Berners-Lee, D. Connolly, *HyperText Markup Language Specification -2.0*, Internet Draft, november 28, 1994.
- [3] V. Quint, I. Vatton, « Grif: an Interactive System for Structured Document Manipulation », *Text Processing and Document Manipulation, Proceedings of the International Conference* (J. C. van Vliet, ed.), 200–213, Cambridge University Press, 1986.
- [4] V. Quint, I. Vatton, « Combining Hypertext and Structured Documents in Grif », *Proceedings of ECHT'92* (D. Lucarella, ed.), 23–32, ACM Press, Milan, December 1992.
- [5] V. Quint, I. Vatton, « Making Structured Documents Active », *Electronic Publishing – Origination, Dissemination and Design*, vol. 7, num. 3, 1994.
- [6] C. M. Sperberg-McQueen, Robert F. Goldstein, « HTML to the Max A Manifesto for Adding SGML Intelligence to the World-Wide Web », <http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Autools/sperberg-mcqueen/sperberg.html>, October 1994.