

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

☞ GRIF ET L'ÉDITION DE DOCUMENT  
STRUCTURÉS ; NOUVEAUX DÉVELOPPEMENTS

☞ Vincent QUINT, Irène VATTON, Jacques ANDRÉ, Hélène RICHY

*Cahiers GUTenberg*, n° 9 (1991), p. 49-65.

<[http://cahiers.gutenberg.eu.org/fitem?id=CG\\_1991\\_\\_9\\_49\\_0](http://cahiers.gutenberg.eu.org/fitem?id=CG_1991__9_49_0)>

© Association GUTenberg, 1991, 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.



# Grif et l'édition de documents structurés : nouveaux développements

---

Vincent QUINT\*, Irène VATTON\*, Jacques ANDRÉ\*\* et Hélène RICHY\*\*

\* *IMAG, 2 rue de Vignate, 38610 Gières*

\*\* *IRISA, Campus de Beaulieu, 35042 Rennes Cedex*

**Résumé.** Grif est un système interactif pour la production et la consultation de documents structurés. Il permet de manipuler des documents complexes comportant des formules mathématiques, des tableaux, des schémas, etc., en mettant l'accent sur l'organisation logique des documents. Cet article présente les principales caractéristiques du système dans son état actuel et fait le point sur les développements en cours.

## 1. Introduction

Grif est un système de production de documents principalement destiné à la documentation scientifique et technique et, d'une façon plus générale, aux domaines où l'on manipule des documents fortement structurés [2][3]. Il est disponible sur station de travail Unix et utilise le système de fenêtres X. A la différence des logiciels de P.A.O., Grif ne s'intéresse pas uniquement à la forme graphique des documents. Ce n'est pas non plus un traitement de texte, bien qu'il puisse effectuer les principales fonctions de ces systèmes. C'est un éditeur structural, dont les concepts de base se comparent plutôt à ceux d'un éditeur syntaxique.

Les premiers travaux qui ont conduit au système Grif ont commencé en 1982, par la conception d'un modèle [4] capable de représenter des documents aussi bien pour un système de gestion de bases de données que pour un éditeur ou pour un formateur. Sur la base de ce modèle un éditeur a été construit dont le premier prototype a été démontrable à partir de 1985 [19]. Ce prototype a ensuite été complété pour donner un outil réellement utilisable en 1988 ; cette version est présentée dans [9]. Parmi les fonctions qui ont été développées durant cette période, figure notamment l'exportation des documents vers des formateurs comme T<sub>E</sub>X [18] ou Mint [1]. Depuis 1988, les principales évolutions ont concerné l'interface utilisateur et les fonctions

de consultation, en particulier les aspects hypertextes. Pendant cette même période, un transfert industriel a donné lieu à un produit commercial<sup>1</sup>.

La section suivante présente l'ensemble du système Grif tel qu'il est en 1991. Après neuf ans de travail, il s'agit d'une étape importante dans le projet, mais pas d'une fin. De nouveaux développements sont en cours, qui permettront de tirer de l'édition structurée des avantages supplémentaires. Ces développements sont présentés dans la section 3.

## 2. Le système Grif

### 2.1. Structure logique des documents

Tous les traitements que Grif peut effectuer s'appuient sur un *modèle de document* de haut niveau. Pour Grif, un document est d'abord une *structure logique*. Ainsi, un article scientifique est considéré comme une suite d'éléments typés : un titre, un ou plusieurs noms d'auteurs, un résumé et une suite de sections, le résumé étant formé de paragraphes et chaque section comportant un titre, quelques paragraphes et une suite de sections de niveau inférieur. Cette organisation convient bien à un article, mais elle ne permet pas de représenter correctement un livre ou une lettre. C'est pourquoi il n'y a pas un modèle unique de document mais un *méta-modèle* qui permet de décrire plusieurs modèles, un pour chaque *classe de documents*. Grâce au méta-modèle, on peut définir une structure logique pour la classe de documents Article, une autre pour la classe de documents Livre, une autre pour la classe de documents Lettre, et d'autres encore, selon les documents à traiter. Il s'agit de *structures logiques génériques*, puisque chacune définit l'organisation de toute une classe de documents. Une structure générique spécifie les types d'éléments qui composent les documents d'une classe et la façon dont ces éléments peuvent s'organiser pour former un document correct. Un langage, appelé S, permet de spécifier des structures logiques génériques sous la forme de *schémas de structure*.

Chaque document traité par Grif possède une *structure logique spécifique*, propre au document, mais conforme à la structure générique de sa classe de documents. Ainsi, plusieurs articles (documents de la classe Article) ont un

---

<sup>1</sup>Le nom Grif est associé à 3 choses différentes : 1) un prototype en cours de développement à l'INRIA dans le cadre du projet Opéra (à Grenoble et Rennes), 2) un produit commercial issu du précédent et 3) la société Grif SA, qui commercialise le produit précédent (Grif SA, 2, bd Vauban, BP 266, 78053 St Quentin en Yvelines Cedex). Nous ne parlons ici que du prototype 1.

nombre différent de noms d'auteurs, de sections ou de paragraphes, mais ils sont bien construits sur le même modèle, celui de la structure logique générique de leur classe de documents.

La structure logique présente de nombreux avantages. Elle permet de produire des documents conformes à un modèle lorsque c'est nécessaire. Elle permet aussi des traitements puissants. En utilisant la structure générique, le système peut garantir que le document a une structure spécifique conforme. Contrairement à la plupart des systèmes de PAO, cette structure est hiérarchique, ce qui permet, par exemple, d'interdire de mettre un résumé au milieu d'une section. En utilisant la structure spécifique, le système peut effectuer des numérotations, manipuler des parties cohérentes du document, constituer des index ou des tables (des matières, des figures...), etc.

## 2.2. Présentation des documents

Un autre intérêt de la structure logique est qu'elle permet de produire automatiquement l'aspect graphique des documents. A chaque type d'élément de la structure logique générique est associé un ensemble de *règles de présentation* qui définissent l'aspect graphique de ce type d'élément. En appliquant ces règles aux éléments de la structure logique spécifique, on peut construire l'image du document. On débarrasse ainsi l'utilisateur du travail de mise en forme du document, on assure une bonne homogénéité de la présentation (tous les éléments de même type sont présentés par application des mêmes règles), et on peut assurer, lorsque c'est nécessaire, que le document est présenté selon un modèle imposé.

Un ensemble de règles de présentation définissant l'aspect graphique de tous les types d'éléments d'une classe de documents est appelé un *schéma de présentation*. Les schémas de présentation sont écrits dans un langage appelé P, qui permet aux utilisateurs de spécifier leurs propres présentations. Ces schémas correspondent à la notion de maquette des typographes. Si on définit plusieurs schémas de présentation pour une classe de documents, en appliquant l'un ou l'autre de ces schémas, on peut voir le même document sous différentes formes graphiques, sans affecter ni le contenu ni la structure logique du document.

Un schéma de structure correspond à un style tel que *article* en  $\LaTeX$  et un schéma de présentation correspond au sous-style *gut.sty*.  $\LaTeX$  ne séparant pas les concepts de structuration logique de ceux de présentation, les langages S et P correspondent au seul langage de macros de  $\LaTeX$ .

### 2.3. Composants structurés

L'approche développée jusqu'ici ne s'applique pas seulement à la structure globale du document, mais également aux différents composants qui interviennent dans un document [21]. Une formule mathématique, un tableau, un schéma, une référence bibliographique ont chacun une structure logique, qui peut être représentée de la même façon que celle d'un document, par une structure générique pour chaque classe de composants et par une structure spécifique pour chaque composant. De même, des règles de présentation peuvent leur être appliquées. Grâce à un méta-modèle suffisamment général et à des règles de présentation bien adaptées, tous ces composants sont représentés de la même façon que l'ensemble du document et ils peuvent donc être traités par les mêmes moyens.

Grif peut éditer plusieurs documents simultanément, qui appartiennent à des classes de documents différentes. De plus, pour éditer des documents incluant des formules, des tables, et toutes sortes de composants, plusieurs structures génériques sont nécessaires : une pour la classe du document et une pour chaque classe de composants. Pour résoudre ce problème, Grif autorise l'importation de structures génériques à l'intérieur d'autres structures génériques. Ainsi un document de la classe Actes de conférence peut contenir des composants des classes Rapport ou Article ; un document Rapport peut contenir des composants des classes Formule ou Tableau et un composant Tableau peut contenir des composants Formule.

### 2.4. Liens entre documents

Les structures logiques des documents sont essentiellement des structures arborescentes. Mais un document ne peut pas, en général, se réduire à un simple arbre. C'est pourquoi le méta-modèle comprend les *références*, qui introduisent des liens non hiérarchiques entre éléments d'arborescences. Une référence peut représenter soit un renvoi (à une note, à une section, à une figure, à la bibliographie, à un index, etc.), soit l'inclusion d'un élément (ou de la totalité) d'un document. Comme il s'agit d'un système interactif, il est plus facile qu'en  $\text{\LaTeX}$  de faire des renvois de plusieurs endroits à la même note ou d'avoir plusieurs systèmes de notes dans une même page.

Toutes les références, les renvois comme les inclusions, peuvent désigner des éléments logiques qui appartiennent au même document que la référence

(référence *interne*) ou à un document différent (référence *externe*), donnant ainsi à Grif un aspect hypertexte.

L'inclusion est une copie vivante, maintenue à jour par le système. Elle est affichée comme une copie de l'élément inclus, mais cette copie n'est pas modifiable. En revanche, son image est mise à jour lors de toute modification effectuée sur l'élément qu'elle copie s'il s'agit d'une inclusion interne, seulement lors de la sauvegarde du document contenant l'élément original s'il s'agit d'une inclusion externe.

Comme tout élément logique, les références sont définies dans les schémas de structure et leur aspect est spécifié dans les schémas de présentation. On peut ainsi définir de nouvelles références, selon les besoins.

## 2.5. Impression et exportation

L'impression des documents peut se faire de deux façons différentes. La forme imprimée peut être engendrée par application des règles de présentation utilisées pendant l'édition ; la sortie sur papier est alors identique à la forme affichée sur l'écran et, dans ce cas, elle est produite en PostScript par Grif. On obtient ainsi un comportement WYSIWYG. Mais on peut aussi sortir sur papier une image différente de celle affichée sur l'écran en utilisant un formateur [1]. Cette seconde option fait intervenir un troisième langage, le langage T, qui permet d'exprimer des règles de traduction, regroupées dans des *schémas de traduction*. Un schéma de traduction spécifie le transcodage des caractères ainsi que des chaînes de caractères à engendrer pour chaque élément d'un document, en fonction de son type et de sa position dans la structure logique du document.

Un programme de traduction fait partie du système. Il lit un document produit par l'éditeur et, en suivant les règles exprimées dans un schéma de traduction, il produit un document traduit. Comme pour les schémas de présentation, on peut définir plusieurs schémas de traduction pour une même classe de documents, ce qui permet de traduire les documents d'une classe dans plusieurs formalismes différents.

La traduction peut être utilisée pour produire des documents acceptables par un formateur comme T<sub>E</sub>X ou L<sup>A</sup>T<sub>E</sub>X [18]<sup>2</sup>. Elle peut aussi être utilisée pour coder les documents selon un standard comme SGML (voir section 3.1).

---

<sup>2</sup>Le présent article a été écrit avec Grif puis traduit en L<sup>A</sup>T<sub>E</sub>X pour être publié dans *Les cahiers GUTenberg*.

D'une façon générale, elle assure qu'un document produit avec Grif pourra être utilisé dans des contextes différents et avec des applications différentes.

## 2.6. Utilisation de Grif

Le rôle de l'éditeur Grif est de créer et de modifier des documents de façon interactive et en suivant les structures logiques génériques. Grâce aux structures génériques, l'éditeur peut guider l'utilisateur dans la construction du document. Ainsi, en reprenant l'exemple donné plus haut, lorsque l'utilisateur demande à créer un article, l'éditeur engendre un titre, un nom d'auteur, un résumé avec un premier paragraphe, une section avec un titre et un paragraphe. Chacun de ces éléments de la structure logique est affiché selon le schéma de présentation choisi. Le titre peut être centré et en gros caractères, le nom d'auteur en italiques, le résumé précédé du mot "Résumé", le titre de section précédé de son numéro, etc. Il ne reste à l'utilisateur qu'à choisir avec la souris lequel de ces éléments vides il veut remplir (les éléments vides sont représentés sur l'écran par un rectangle gris) et à frapper le texte au clavier. Chaque fois qu'il veut créer un nouvel élément structural, l'utilisateur frappe une touche de commande ou appelle un menu qui lui donne le choix des éléments qu'il peut créer à la position courante, en fonction de la structure logique générique. Là encore, l'éditeur crée les différents constituants de l'élément choisi et les affiche selon le schéma de présentation.

Pendant l'édition d'un document, l'éditeur crée et met à jour une structure interne qui matérialise l'organisation logique du document en cours de création ou de modification. C'est sur cette structure interne, nommée *arbre abstrait*, que l'éditeur effectue tous les traitements ayant trait à la structure : création, destruction, recherche, déplacement, etc.

L'arbre abstrait et la *représentation pivot*, qui en est la traduction sous une forme linéaire plus adaptée au stockage et à la transmission, décrivent la structure logique spécifique du document ainsi que son contenu. Toutes les autres formes de représentation du document sont construites à partir de cette représentation. Pour éviter de figer la forme graphique des documents, ni la représentation pivot ni l'arbre abstrait ne contiennent d'informations de présentation. Mais, à partir de l'arbre abstrait du document, et guidé par les schémas de présentation choisis, l'éditeur engendre une représentation formatée du document, à travers laquelle l'utilisateur interagit avec le système.

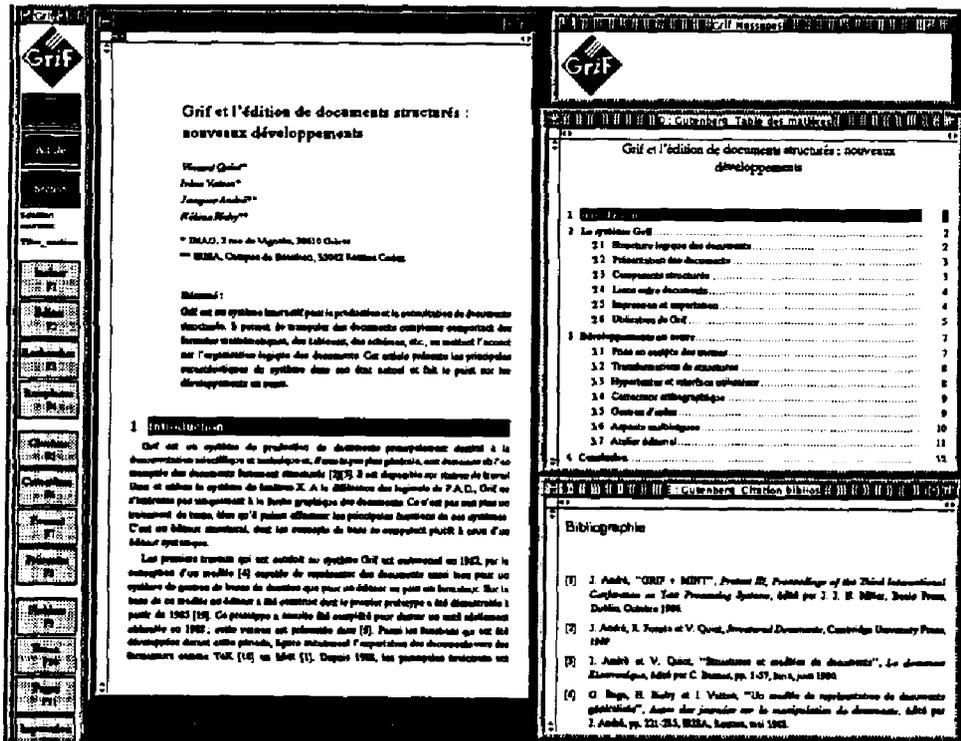


Figure 1. Les vues Grif : vue principale, vue table des matières et vue des références bibliographiques.

L'image présentée à l'écran est essentiellement un moyen de traduire la structure logique spécifique du document en cours de traitement. Pour cette raison, plusieurs vues du même document peuvent être affichées en même temps, dans des fenêtres différentes. Les vues sont définies, pour chaque classe de documents, dans les schémas de présentation. Elles permettent d'afficher certaines parties du document qui présentent un intérêt particulier pour certains traitements. On peut les considérer comme un filtre appliqué sur le document au moment de l'affichage.

Les vues ne constituent pas seulement un moyen de visualisation, elles sont aussi un moyen d'interaction avec le document : toutes les vues peuvent être utilisées de la même façon pour éditer un document. On peut travailler sur la vue table des matières pour créer ou modifier le plan d'un article. Il est alors facile de déplacer une section complète, avec son contenu. La table des matières peut aussi être utilisée pour parcourir rapidement le document. On peut voir et modifier une référence bibliographique dans la vue bibliographie

en même temps qu'on voit, dans une autre vue, la partie du document qui s'y rapporte. Toute modification opérée dans une vue est immédiatement reflétée dans toutes les autres vues. En particulier, les numéros affectés par une opération sont mis à jour dans toutes les vues, y compris les renvois comme "voir section  $n$ " ou "voir figure  $n$ ".

Plusieurs documents peuvent être traités en même temps par l'éditeur Grif. Lorsque de nombreux documents sont nécessaires simultanément, par exemple pour établir des liens entre eux, certains peuvent être chargés sous forme de *squelette*, de façon que seule la partie réellement utile soit présente. Le squelette est défini dans le schéma de structure de la classe de documents ; celui-ci indique les types des éléments à charger dans la forme squelette. Ainsi, dans un article, le squelette ne comprend que le titre de l'article, le résumé complet et les sections de différents niveaux avec seulement leur titre. De nombreux articles peuvent alors être chargés simultanément.

La structure logique d'un document ne fournit pas seulement une aide pour la création ou la visualisation du document, elle augmente aussi la puissance des commandes habituelles de manipulation du document. Ainsi les recherches peuvent s'appuyer sur la structure logique. L'utilisateur peut, par exemple, rechercher un élément d'un type donné (sauter au prochain titre de section, ou à la prochaine formule) ; il peut aussi chercher tous les renvois à un élément donné, ou les copies d'un élément (chercher tous les renvois à cette figure, tous les renvois à cette citation bibliographique) ; il peut au contraire chercher l'élément désigné par un renvoi, ou l'original d'un élément inclus. Grâce à la richesse de la structure logique, on obtient des fonctions analogues à celles des systèmes d'hypertexte [5].

Comme avec  $\text{\LaTeX}$ , il y a au moins deux catégories d'utilisateurs. Les schémas de structure, de présentation et de traduction sont préparés par des spécialistes, qui maîtrisent les langages S, P et T, qui ont les compétences requises pour analyser les classes de documents et pour définir la mise en page et la typographie de ces documents. Il y a d'autre part les utilisateurs finals, qui utilisent les schémas préparés par les précédents et qui n'ont pas besoin du même niveau de compétence.

Grif se présente comme un système générique qui peut être adapté, grâce aux schémas de structure et de présentation, au traitement de documents et de composants très variés. Il tire de là une grande souplesse. L'uniformité des structures sur lesquelles il travaille rend les manipulations à la fois simples et homogènes. Bien que toutes sortes de composants puissent être

traités, l'utilisateur n'a pas affaire à une collection d'éditeurs spécialisés, mais à un environnement unique et cohérent. Le haut niveau d'abstraction du modèle de document confère à l'éditeur une puissance de traitement qu'on ne rencontre pas habituellement dans les systèmes interactifs de production de documents. C'est aussi un système très ouvert : un document saisi avec Grif peut ensuite être traité par d'autres systèmes.

### 3. Développements en cours

L'ensemble des fonctions offertes par Grif permet de traiter des documents complexes dans de bonnes conditions, mais ses caractéristiques principales peuvent encore être exploitées pour offrir de nouveaux services. C'est notamment pour mieux comprendre tous les avantages qu'offrent à un système interactif une forte structuration et une description générique des documents que de nouveaux développements ont été entrepris récemment.

Dans cette section, on présente les évolutions en cours dans Grif. Il s'agit de rendre Grif conforme aux normes internationales de représentation des documents, d'effectuer des traitements plus poussés des structures logiques, de développer ou d'enrichir des structures plus complexes que les simples structures hiérarchiques (index, hypertexte), de s'intéresser davantage au contenu textuel des documents (correction orthographique, multilinguisme) et d'ouvrir l'édition structurée à de nouvelles applications.

#### 3.1. Prise en compte des normes

Le modèle de document qui est à la base de Grif est conceptuellement proche du modèle des normes ODA [12] ou SGML [11]. Dans ces deux normes on a des structures logiques et physiques, des structures génériques et des structures spécifiques. Dans ses principes, Grif est donc bien adapté au traitement de documents représentés de façon normalisée, mais pour exploiter complètement les ressources des normes, il faut aller plus loin.

Un premier travail a été fait pour SGML. Les concepts de SGML qui n'avaient pas d'équivalent dans Grif y ont été introduits. C'est par exemple le cas des exceptions ou de certains types d'attributs. De cette façon, un document préparé avec Grif peut tirer parti de (presque) toutes les possibilités offertes par SGML et, inversement, n'importe quel document SGML peut être traité par Grif. Cela suppose néanmoins que les problèmes d'ordre syntaxique soient résolus (la syntaxe de la forme pivot de Grif n'est

pas SGML). La solution est offerte par le traducteur de Grif, qui produit les sorties en SGML et par un *parser* SGML qui transforme un document SGML en fichier pivot Grif. De cette façon Grif se présente comme un système SGML.

SGML est utilisé pour représenter la structure logique d'un document, mais Grif produit aussi l'aspect graphique des documents et il utilise pour cela le langage P. De même que SGML peut remplacer le langage S, le langage DSSSL [13], actuellement en cours de normalisation, pourra être utilisé pour remplacer le langage P de Grif.

### 3.2. Transformations de structures

Le principe d'un document représenté sous la forme de sa structure logique et conforme à une structure générique a de nombreux avantages, mais il présente aussi quelques difficultés qui ne sont pas toutes résolues. Il s'agit dans la plupart des cas de problèmes de transformation de structures logiques [10].

Ainsi, lorsqu'on déplace (ou qu'on copie) une partie de document à l'intérieur d'un même document ou entre documents différents, il se peut que la structure de cette partie ne soit pas conforme à la structure générique décrivant l'endroit où on veut l'insérer. Une restructuration est alors nécessaire.

Un autre cas de restructuration est la modification des structures génériques : les documents déjà construits selon l'ancienne version de la structure générique doivent être transformés pour se conformer à la nouvelle version.

Un dernier cas de transformation de structure est celui présenté par la structuration *a posteriori* d'un texte saisi sans structure ou encore la restructuration par l'utilisateur d'un texte mal structuré. Ces opérations doivent permettre à un auteur qui écrit un texte d'en modifier la structure à tout moment et de façon confortable. C'est aussi le moyen de permettre une construction ascendante d'un document, alors que l'édition structurée a une forte tendance à favoriser la construction descendante, qui n'est pas toujours la plus naturelle pour un auteur.

### 3.3. Hypertextes et interface utilisateur

On a vu (section 2.4) qu'il existe déjà des liens de nature hypertexte dans Grif, mais ils sont encore peu développés. En particulier, il n'y a que deux types (renvoi et inclusion) et il s'agit de liens passifs : la seule action possible sur les liens est de les suivre, c'est à dire de faire afficher par le système la partie de document qui est la cible du lien, cette cible étant toujours dans un document Grif. Cela offre déjà un moyen de parcours rapide d'un document ou d'un ensemble de documents, en plus du parcours séquentiel ou hiérarchique.

Les liens existant seront généralisés pour permettre de lier aux documents des objets qui ne sont pas des documents (bases de données ou d'objets, programmes, fichiers, etc.) et aussi pour associer librement des traitements aux liens. L'association d'actions à des liens, ou plus généralement d'activités à des documents permettra notamment d'utiliser l'éditeur Grif comme un moyen d'interface utilisateur pour un ensemble d'applications [20].

Avec ce type d'extension, Grif se présentera comme un véritable système d'hypertexte, tout en ayant aussi tous les traits d'un éditeur structuré. On obtiendra ainsi une intégration des fonctions des deux types de systèmes et on pourra voir les hypertextes à travers l'approche générique.

### 3.4. Correcteur orthographique

Un correcteur orthographique a été intégré à Grif. Ce correcteur, écrit à l'Irisa [8], est basé sur l'utilisation de dictionnaires et sur l'algorithme de Levenshtein [14]. Lorsqu'il détecte un mot inconnu, il suggère des corrections possibles.

L'utilisateur de Grif décide, s'il s'agit d'une erreur, de corriger le mot sélectionné par le correcteur. S'il ne s'agit pas d'une erreur, le mot peut être ajouté dans un dictionnaire qui est sauvegardé en même temps que le document.

La recherche des mots à corriger parcourt *tous* les éléments textuels de l'arbre du document, qu'ils soient dans des paragraphes, des tableaux ou des graphiques. La substitution du mot erroné par le mot corrigé est effectuée immédiatement dans le document. La notion de *langue* (représentée par un attribut Grif) permet au correcteur de travailler sur des documents écrits dans différentes langues en utilisant les dictionnaires correspondant, ou d'ignorer les textes portant l'attribut d'une *langue* pour laquelle le

dictionnaire n'est pas disponible. Ainsi des mots inconnus des dictionnaires pourront-ils être maintenus dans le document sans être détectés comme erronés lors de chaque utilisation du correcteur.

De même, un correcteur de micro-typographie (espacement accompagnant les signes de ponctuation, emploi des capitales, etc.) est à l'étude.

### 3.5. Gestion d'index

L'index a une fonction importante dans de nombreuses catégories d'ouvrages. Il est indispensable dans les publications scientifiques ou techniques qui sont utilisées comme documentation de travail. Mais, avec  $\LaTeX$ , la construction de tables d'index n'est pas aisée [15].

L'index est également très utile pour manipuler des documents électroniques : en désignant une entrée de l'index qui lui paraît répondre à sa recherche, l'utilisateur peut visualiser directement la partie du document qui l'intéresse. Mais les systèmes de P.A.O. qui produisent des index ne les utilisent généralement pas comme des liens permettant de parcourir les documents électroniques.

L'introduction d'une fonction de tri va permettre à Grif de construire des tables d'index. A partir des définitions de termes à indexer et de leur localisation dans un document, il sera possible de construire et de présenter des listes ordonnées des termes et de leur références. Interactivement, l'utilisateur pourra suivre les liens (références internes) entre le document et l'index, ajouter des termes nouveaux ou modifier des termes existants afin d'améliorer le contenu de l'index, Grif se chargeant d'effectuer le tri des entrées et le calcul des références aux pages (lorsque la pagination change). Comme pour n'importe quel document Grif, la présentation des tables d'index est définie par le schéma de présentation choisi.

L'exploitation des propriétés structurales des documents permet de définir des zones d'indexation privilégiées dans lesquelles seront définis automatiquement des termes de l'index. Par exemple, dans un document de la classe Article, les noms des auteurs et les mots-clés sont des éléments structurés qui contiennent des termes susceptibles d'être retenus comme termes d'indexation. Ainsi, quel que soit le document de cette classe, l'index initial contiendra les noms extraits de la liste des auteurs et de la liste des mots-clés.

L'utilisation des références externes permettra également de construire des index multi-documents.

### **3.6. Aspects multilingues**

Les problèmes de multilinguisme concernent un système comme Grif de différentes façons :

- Plusieurs langues doivent pouvoir être utilisées à l'intérieur du même document, en respectant les usages linguistiques et typographiques de chaque langue (par exemple, imprimer du français de gauche à droite et de l'arabe de droite à gauche ou couper structure dans un passage en français et structure dans un passage en anglais).
- Le dialogue entre le système et l'utilisateur doit pouvoir se faire dans une autre langue que celles utilisées dans le texte des documents édités.
- La même souplesse doit être offerte aux utilisateurs finals dans le choix des langues que dans le choix des types de documents ; en particulier un utilisateur doit pouvoir ajouter une nouvelle langue dans le système.

Au moins quatre séries de problèmes se posent dans Grif pour atteindre ces objectifs :

**Codage** : le codage des caractères sur huit bits (utilisé par l'alphabet ISO-Latin 1) n'est pas suffisant pour toutes les langues.

**Saisie** : les claviers adaptés à une langue particulière ne permettent pas de traiter efficacement plusieurs langues à la fois. De plus les claviers ne sont bien adaptés qu'aux langues alphabétiques ; il reste le problème des langues idéographiques.

**Restitution** : de nombreuses polices de caractères sont nécessaires. Ces polices peuvent être très volumineuses pour des langues comme le chinois.

**Dialogue** : le dialogue de l'application est complexe puisqu'il fait intervenir des éléments qui ne dépendent que du système, d'autres qui viennent des schémas de structure utilisés (types des composants logiques disponibles, par exemple) et d'autres encore qui sont extraits des documents eux-mêmes. Pour proposer un dialogue cohérent, tous ces

éléments doivent être présentés dans la même langue, choisie par l'utilisateur.

L<sup>A</sup>T<sub>E</sub>X sait traiter certains de ces problèmes, mais pas tous en même temps. Dans Grif, une première approche du problème a été faite à Lille et a été validée pour le vietnamien et le chinois [16]. Cette méthode est fondée sur un nouveau langage qui permet de décrire les spécificités d'une langue (méthodes de codage, de saisie, de restitution, éléments de dialogues), ce qui confère au système une certaine souplesse et de réelles possibilités d'évolution dans le domaine du multilinguisme.

### 3.7. Atelier éditorial

Un éditeur, aussi puissant soit-il, ne résoud qu'une partie du problème plus général du traitement des documents, surtout lorsqu'il s'agit de documents complexes, comme ceux de la documentation technique ; d'autres outils sont nécessaires. On peut trouver de nombreux points communs entre le domaine de la documentation et celui du génie logiciel [17]. Dans cette optique, on peut parler de *génie éditorial* et d'*atelier éditorial* comme on parle de génie logiciel et d'atelier logiciel. De même qu'un éditeur de texte ou un éditeur syntaxique ne suffit pas pour développer de gros programmes, de même un éditeur de documents, même bien structuré, ne suffit pas au traitement des documents complexes.

Un atelier éditorial peut être vu comme un environnement de développement de documents, sensiblement comparable à celui fourni par un atelier logiciel, mais orienté vers la production de documents plutôt que de programmes [22]. Un atelier éditorial doit permettre d'éditer des documents, de les imprimer, de gérer des versions, de les stocker, de les retrouver, de produire des configurations adaptées et de maintenir des liens. Il doit aider au suivi et à la coordination du travail d'un ensemble d'utilisateurs (auteurs et lecteurs) sur un ensemble de documents. C'est typiquement une application de travail coopératif [7]. L'atelier éditorial doit veiller à l'intégrité de l'information contenue dans les documents et partager l'accès aux documents entre les utilisateurs.

Le développement d'un tel atelier est un travail considérable et notre objectif, au moins à court terme, n'est pas de développer un atelier complet, mais plutôt d'étudier quelques problèmes posés par ce genre d'environnement, en développant une application couvrant certains des aspects d'un atelier éditorial.

La première application qui a été développée permet à plusieurs utilisateurs, sur des stations de travail différentes, de partager l'accès à un ensemble de documents. Cette application a été développée sur Guide [6] qui est un système d'exploitation distribué à objets. Guide prend en charge les problèmes de stockage des objets (les documents et les différents schémas) et offre les mécanismes de répartition, de communication et de synchronisation nécessaires. L'éditeur Grif a été porté sur Guide et est disponible sur les stations de travail contrôlées par le système Guide. L'application, qui utilise les services de Guide et l'ensemble des fonctions de Grif, permet aux utilisateurs d'accéder simultanément aux mêmes documents, en découpant chaque document en parties, chacune de ces parties ne pouvant être accédée en écriture que par un utilisateur à la fois. Lorsqu'un utilisateur modifie le contenu ou la structure d'une de ces parties, les autres utilisateurs qui accèdent au même document à cet instant voient sur leur écran les modifications effectuées par le premier.

D'autres aspects d'un atelier éditorial mériteraient d'être étudiés et ne le sont pas encore. Deux principalement paraissent importants :

- La gestion des versions de documents est nécessaire dans un projet éditorial. Les outils de génie logiciel qui gèrent des programmes en conservant une version de référence et les différences par rapport à cette version ne peuvent pas être appliqués directement à des documents structurés. Les différences devraient au moins être exprimées par rapport à la structure logique du document et il est donc nécessaire de savoir comparer et fusionner des structures de documents.
- L'architecture classique d'un éditeur ne convient pas bien à de grosses applications éditoriales, où l'édition de documents ne constitue qu'une des fonctions de l'ensemble. Les fonctions d'édition doivent pouvoir être intégrées avec les autres composants d'une façon plus intime que par des communications de fichiers. Un nouveau type d'architecture logicielle est nécessaire.

#### 4. Conclusion

Dans cet article, nous avons tenté de faire le point sur l'état actuel du système Grif et sur ses développements les plus récents. Il apparaît qu'il ne s'agit là que d'une étape et que, bien que le système offre des services

puissants, il peut encore être utilisé comme base expérimentale pour étudier de nombreux problèmes qui restent posés dans le champ des documents électroniques, et plus particulièrement dans les documents structurés.

Parmi ces problèmes, on peut en indiquer certains qui devraient être traités dans le prolongement du projet Grif. Il y a d'abord les problèmes qui sont présentés dans la section 3 et dont l'étude se poursuit : transformations de structures, hypertextes, correction typographique, indexation, multilinguisme, applications éditoriales (gestion de versions notamment). Il y a aussi des problèmes qui ne sont pas encore abordés, comme les problèmes d'architecture logicielle et de présentation graphique.

## Références bibliographiques

- [1] J. André, "GRIF + MINT", *Protext III, Proceedings of the Third International Conference on Text Processing Systems*, J. J. H. Miller, ed., Boole Press, Dublin, Octobre 1986.
- [2] J. André, R. Furuta, V. Quint, *Structured Documents*, Cambridge University Press, 1989.
- [3] J. André, V. Quint, "Structures et modèles de documents", *Le document Electronique*, C. Bornes, ed., pp. 1-57, Inria, juin 1990.
- [4] G. Bogo, H. Richy, I. Vatton, "Un modèle de représentation de documents généralisés", *Actes des journées sur la manipulation de documents*, J. André, ed., pp. 221-235, IRISA, Rennes, mai 1983.
- [5] J. Conklin, "Hypertext : An Introduction and Survey", *Computer*, vol. 18, num. 9, pp. 17-41, 1987.
- [6] D. Decouchant, M. Riveill, C. Horn, E. Finn, "Experience with implementing and using an object-oriented, distributed system", *Proceedings of Usenix Workshop on Experiences with Distributed and Multiprocessor Systems (Fort Lauderdale)*, pp. 301-310, October 1989.
- [7] C. A. Ellis, S. J. Gibbs, G. L. Rein, "Groupware, some issues and experiences", *Communications of the ACM*, vol. 34, num. 1, pp. 38-58, January 1991.
- [8] P. Frison, D. Lavenier, "A fast machine for prototyping string correction algorithms", *International Conference on the user-oriented content-based text and image handling, RIAO88*, Cambridge MIT, Mars 1988.
- [9] R. Furuta, V. Quint, J. André, "Interactively Editing Structured Documents", *Electronic Publishing - Origination, Dissemination and Design*, vol. 1, num. 1, pp. 19-44, Avril 1988.

- [10] R. Furuta, P. David Stotts, "Specifying Structured Document Transformations", *Document Manipulation and Typography*, J. C. van Vliet, ed., pp. 109 - 120, Cambridge University Press, 1988.
- [11] I.S.O., *Information processing - Text and office systems - Standard Generalized Markup Language (SGML)*, ISO 8879, October 1986.
- [12] I.S.O., *Information processing - Text and office systems - Office Document Architecture (ODA)*, ISO 8613, 1989.
- [13] I.S.O., *Information technology - Text and office systems - Document Style Semantics and Specification Language (DSSSL)*, ISO/IEC DIS 10179, 1991.
- [14] V.I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals ", *Sov. Phys. Dokl.*, vol. 10, num. 8, pp. 707-710, Février 1966.
- [15] Ph. Louarn, "Traitement d'index avec L<sup>A</sup>T<sub>E</sub>X", *Cahiers GUTenberg*, num. 7, pp. 23-28, Novembre 1990.
- [16] Phan Huy Khán, *Contribution à l'informatique multilingue, extension d'un éditeur de documents structurés*, thèse, Université des Sciences et Techniques de Lille Flandres Artois, mai 1991.
- [17] V. Quint, M. Nanard, J. André, "Towards Document Engineering", *EP'90*, R. Furuta, ed., pp. 17-29, Cambridge University Press, Septembre 1990.
- [18] V. Quint, I. Vatton, H. Bedor, "Grif, an Interactive Environment for T<sub>E</sub>X", *T<sub>E</sub>X for Scientific Documentation*, J. Désarménien, ed., pp. 145-158, Springer-Verlag, 1986.
- [19] 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., pp. 200-213, Cambridge University Press, 1986.
- [20] V. Quint, I. Vatton, "Une boîte à outils pour les images", *Colloque sur l'Ingénierie des Interfaces Homme-Machine*, pp. 17-28, GIP-Altair, INRIA, Sophia-Antipolis, mai 1989.
- [21] V. Quint, I. Vatton, "Modularity in structured documents", *Woodman'89*, J. André J. Bézivin, ed., pp. 170-177, Bigre num. 63-64, IRISA, Rennes, mai 1989.
- [22] J. H. Walker, "Supporting Document Development with Concordia", *Computer*, vol. 21, num. 1, pp. 48-59, January 1988.