Techniques d'interaction et de visualisation pour l'accès à des documents numérisés

Laurent Robert, Eric Lecolinet
Ecole Nationale Supérieure des Télécommunications / CNRS URA 820
46 rue Barrault, 75013 Paris, France
{lrobert,elc}@enst.fr – http://www.enst.fr/{~lrobert,~elc}

RESUME
Les systèmes hypermédia sont amenés à traiter de vastes quantités de données, dont un nombre de plus en plus important de documents numérisés, stockés sous forme d'images. Cet article décrit des techniques d'interaction homme-machine et de visualisation d'information développées pour faciliter l'utilisation et l'accès de tels documents. On décrit tout d'abord une procédure interactive utilisée pour annoter des images de textes manuscrits en vue de leur indexation. Cette procédure permet un codage XML du contenu de l'image et offre des possibilités étendues de couplage entre ces deux représentations (texte et image). Nous proposons ensuite plusieurs techniques de visualisation destinées à faciliter le repérage, la localisation et la navigation dans un espace hypermédia.

MOTS CLES : hypermédia, documents numérisés, annotation, couplage texte/image, édition et codage XML, visualisation d'information, exploration et navigation, espace de travail, représentation multi-échelles, mémoire visuelle.

INTRODUCTION
De nombreux documents sont aujourd'hui disponibles sous forme numérisée (photographies, plans et cartes, rapports, notes et autres documents manuscrits, journaux, etc.) [5]. Cependant, accéder à ces documents au sein d'un espace documentaire de vaste envergure peut s'avérer difficile, et ce quelle que soit la technique utilisée (indexation ou exploration interactive).

Le premier problème rencontré est lié à la nature de ces documents numérisés : ce sont des images bitmaps qu'il est difficile d'exploiter informatiquement. Pour faciliter la recherche d'information, une solution consiste à annoter les images pour associer à certaines de leurs parties des mots clés, des notes ou plus généralement une représentation textuelle de leur contenu. L'indexation et d'autres traitements informatiques peuvent alors s'effectuer facilement sur ces représentations textuelles. Il est également possible, par ces techniques d'annotation, d'associer à une image de la documentation connexe (d'autres images, du texte, du son, etc.) par l'intermédiaire d'hyperliens.

L'accès aux documents par exploration interactive de l'espace informationnel pose également certains problèmes. En effet, les utilisateurs ont souvent la sensation " d'être perdus " dans la masse de données présentées. Ceci entraîne d'une part des problèmes de localisation (d'où vient-on ? quels liens ont été suivis ? où se trouve un document préalablement consulté ?) et d'autre part, des difficultés à effectuer des associations d'idées (par la difficulté d'afficher simultanément plusieurs composants ainsi que le chemin qui les relie). De plus, les utilisateurs sont souvent perturbés par le nombre important de choix de navigation qui leur sont proposés. Ils sont confrontés au syndrome de déso-rientation dans l'hyperespace [12] dont les principales sources sont : l'importante quantité de documents manipulés et la complexité du réseau de liens.

Dans cet article, nous proposons quelques solutions à cette double problématique d'accès à des documents numérisés (par le contenu et par navigation). Comme exemple pratique, les travaux effectués ont été appliqués, dans le cadre du projet Philectre [9], à un espace hypermédia contenant des images de textes manuscrits littéraires, dont un exemple est donné à la Figure 1. Dans le cadre de cette étude, un de nos objectifs consiste à fournir aux chercheurs en littérature des outils pour faciliter l'exploitation informatique d'images de manuscrits (codage de leur contenu, accès, navigation et organisation interactive, etc.).

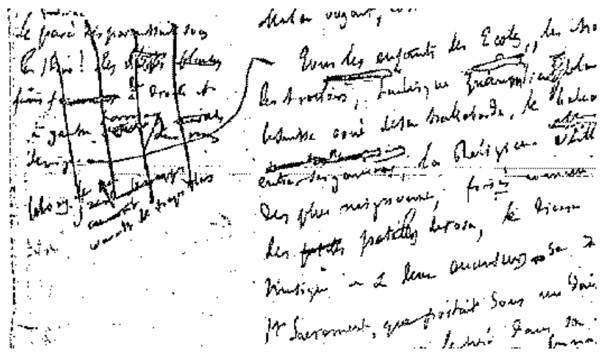

Figure 1: Image d'un texte manuscrit littéraire (Flaubert).

La prochaine section décrit un système qui permet, par annotation interactive, de mettre en correspondance des parties d'une image avec du texte structuré codé au format XML [18]. Une attention particulière est consacrée à l'association texte/image correspondante et au codage des informations nécessaires à un tel couplage. La représentation visuelle du code XML produit s'effectue quant à elle par l'intermédiaire d'un nouveau toolkit graphique. Dans la section suivante sont présentées des techniques qui permettent l'indexation visuelle de documents et la navigation en contexte, ceci afin de faciliter l'exploration interactive de l'espace informationnel.

COUPLAGE TEXTE/IMAGE

Edition et Couplage Interactif

L'objectif principal consiste à annoter, de manière interactive et intuitive, des parties d'images en vue de pouvoir les indexer via les annotations produites. Ces annotations sont codées en XML afin de faciliter leur traitement informatique.

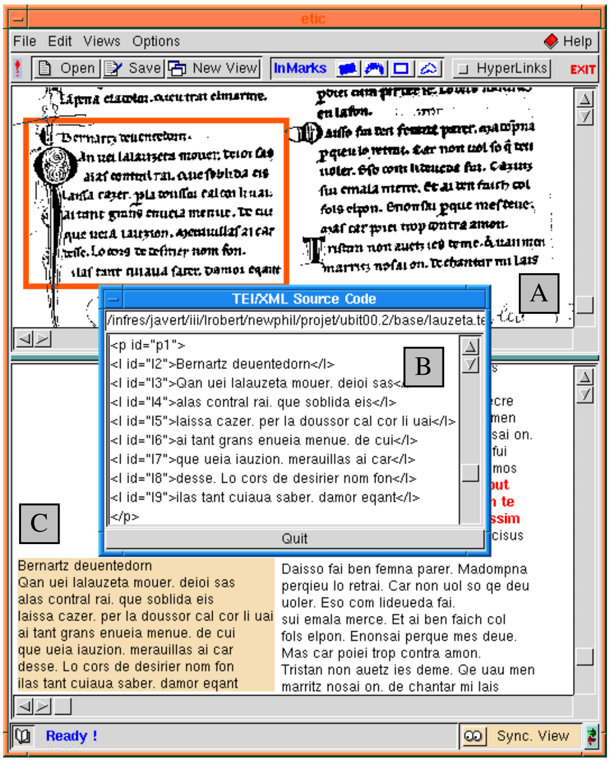

Figure 2: L'outil d'édition et de consultation. A: image du manuscrit. B: code XML. C: représentation graphique du code XML.

Pour l'exploitation de notre base documentaire littéraire, nous avons développé un système qui permet de transcrire (sous forme de texte structuré XML) le texte contenu dans une image de manuscrit et d'effectuer le couplage texte/image correspondant. L'image est interactivement découpée en plusieurs parties. Chacune d'elle est ensuite associée à une balise XML. Ceci peut être vu comme une tentative d'unification des représentations textuelles internes (le codage XML) et externe (l'image) en fournissant une correspondance biunivoque entre ces représentations. Ce schéma facilite également l'accès par le contenu aux images en permettant de référencer des parties d'images par l'intermédiaire de leurs liens au texte structuré. Notre système, illustré à la Figure 2, peut être utilisé à la fois pour l'édition et pour la consultation de matériaux préalablement édités.

Procédure de couplage. Pour désigner une zone particulière à l'intérieur d'une image, on propose plusieurs outils interactifs (des marqueurs). Ces marqueurs permettent de déposer une trace graphique à l'aide de la souris sur ou autour d'une portion d'image, comme illustré à la Figure 3. Quatre types de marqueurs ont été développés (ligne, stabilo, rectangle, lasso).

La phase de transcription (édition d'une version XML) s'effectue en deux étapes successives. L'utilisateur doit d'abord sélectionner une zone de texte manuscrit dans l'image (à l'aide d'un marqueur interactif) et ensuite taper le texte correspondant.

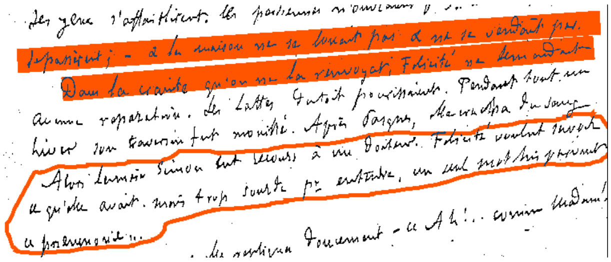

Figure 3: Marqueurs interactifs, dépôt de traces graphiques dans l'image pour sélectionner des parties.

Une fois la zone de texte sélectionnée, une interface de transcription apparaît sur l'image du manuscrit à côté de cette zone. L'utilisateur y entre la transcription du texte manuscrit, et choisit parmi un ensemble de balises (ligne, paragraphe, etc.) celle qui sera associée à cette partie du texte. Après validation, un élément XML est automatiquement créé et ajouté à la représentation interne du document XML.

La proximité spatiale entre le texte manuscrit et la zone d'édition permet à l'utilisateur de se concentrer sur la tâche de transcription (interprétation et édition du texte), en lui évitant d'avoir à se concentrer sur deux parties différentes (voire éloignées) de l'écran. Comme le montre la figure 4, cette interface est composée d'un objet graphique transparent qui augmente cet effet de proximité visuelle. L'utilisateur a l'impression d'écrire sur l'image et peut ainsi transcrire le texte en contexte. Cet effet de transparence peut être désactivé au besoin, si la superposition du texte manuscrit et du texte entré au clavier rend leur lisibilité difficile.

La correspondance entre la zone image sélectionnée et le texte (codé en XML) correspondant est alors effectuée automatiquement, un lien bidirectionnel étant implicitement créé. Cette équivalence est matérialisée par un surlignage dans les deux zones en correspondance chaque fois que l'on sélectionne l'une d'entre elles (Figure 2, partie a et c). Enfin, le système préserve la disposition spatiale des éléments transcrits: les coordonnées du polygone sélectionné dans l'image sont sauvegardées tandis que le texte est automatiquement ajusté dans la vue graphique (Figure 2, partie c) de manière à ce qu'il soit approximativement disposé comme dans l'image. Ceci dans le but d'aider l'utilisateur à effectuer la correspondance visuelle entre image et texte.

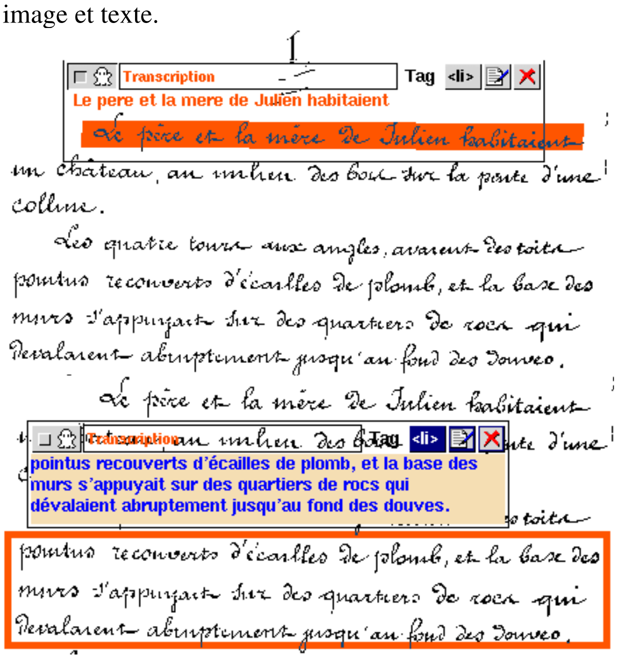

Figure 4: L'outil de transcription transparent (en haut). La transparence peut être désactivée au besoin (en bas).

Du fait de la masse de documents qu'il faut généralement traiter, il serait vite fastidieux d'avoir à établir cette tâche de mise en correspondance de manière purement manuelle, même si la procédure interactive précédemment décrite peut en faciliter la réalisation. Nous proposons donc une approche semi-automatique laissant à l'utilisateur la possibilité de valider ou de corriger interactivement des hypothèses de segmentation qui lui sont fournies par un module d'analyse automatique de document, décrit plus précisément dans [9].

Couplage Texte/Image en XML/TEI

L'utilisation d'un format de codage standard comme XML offre plusieurs avantages. Tout d'abord, il facilite le partage et la réutilisation des données. Le traitement et l'interprétation de ces données peuvent être spécifiques à une application et s'effectuer à plusieurs niveaux de granularité, en appliquant par exemple des “ filtres ” sur les balises.

Un ensemble de modules séparés, partageant les ressources XML, peuvent être développés, chacun prenant en charge une tâche spécifique. L'ajout ou l'évolution de modules ne remet ainsi pas en cause l'architecture du système. Ceci permet aux utilisateurs de choisir l'outil le plus approprié à leur besoin. Ils ne sont plus contraints d'utiliser un logiciel unique pouvant rapidement devenir obsolète ou n'étant pas assez flexible pour réaliser un ensemble varié de tâches données. Enfin, ce type de codage peut grandement simplifier et améliorer des traitements informatiques ultérieurs tels que la création de tables d'index, le calcul de statistiques ou l'utilisation de moteurs de recherche. En effet, en plus du contenu du document (le texte), les balises XML permettent de stocker un ensemble varié d'informations sur lesquelles il est possible d'effectuer divers recherches et traitements.

Pour le codage de manuscrits littéraires, nous avons choisis le format XML/TEI [17] qui est un modèle de document spécifiquement utilisé dans le domaine des sciences humaines. Ce format offre une puissance d'expression de haut niveau pour décrire des phénomènes para-textuels (insertions, corrections, textes illisibles, etc.). Il fournit une bonne base à la description d'images de textes manuscrits littéraires et s'adapte donc bien à nos besoins.

Implémentation. Le contenu textuel d'une image de manuscrit est décrit au moyen de balises XML/TEI. On repère par exemple les lignes, les paragraphes et les colonnes par les balises XML/TEI *<lb>*, *<p>* et *<cb>*. A chaque élément *<lb>* et *<p>* est associé un attribut *id* qui sert à les différencier. Cet identifiant nous permet de référencer un élément spécifique afin de l'associer à une partie de l'image (Figure 2).

```
<text><body><cb>
<lb id="l1">Les yeux s'affaiblirent, les persiennes n'ouvraient plus. Bien des années</lb>
<lb id="l2">se passèrent; - & la maison ne se louait pas & ne se vendait pas.</lb>
<p id="p1">
<lb id="l3">Dans la <emph>crainte qu'on ne la renvoya, Félicité </emph>ne demandait</lb>
<lb id="l4">aucune réparation. Les Lattes du toit pourrissaient. Pendant tout un</lb>
<lb id="l5">hiver son traversin fut mouillé. Apres Pâques, elle cracha du sang</lb>
<lb id="l6">Alors Lemaire Simon eut recours a un docteur. Félicité voulut savoir</lb>
<lb id="l7">ce qu'elle avait, mais trop sourte pr entendre, un seul mot lui parvint</lb>
<lb id="l8">"pneumonie"</lb>
</p>
</cb></body></text>
```

Le codage des informations nécessaires au couplage texte/image s'avère un peu plus délicat. En effet, en raison de la nouveauté de XML, les mécanismes de pointage disponibles (encore en cours de définition) ne permettent pas d'indexer une partie d'image de manière standard. Pour pouvoir associer une balise à une portion d'image (un polygone quelconque contenant un ensemble de pixel), nous proposons donc une extension pour un mécanisme de pointage XML/TEI.

Cette mise en correspondance texte/image s'effectue en deux étapes. On crée tout d'abord un pointeur (une balise XML/TEI de nom *<xptr>*) vers chaque zone de l'image que l'on veut indexer. C'est la sémantique de cette balise *<xptr>* qu'il nous a fallu préciser. L'attribut *doc* permet d'indiquer le nom du fichier image traité. L'attribut *type* permet de savoir lequel de nos marqueurs interactifs a été utilisé pour définir la zone image pointée. Les coordonnées de cette zone sont stockées dans deux listes de l'attribut *from*, l'une pour les abscisses, l'autre pour les ordonnées.

```
<ptrGrp>
<xptr id="xp1" type="line" doc="coeur.gif"
        from="space (3d) (0 32 765) (30 208 172)" />
<xptr id="xp2" type="line" doc="coeur.gif"
        from="space (3d) (0 25 753) (30 255 207)" />
<xptr id="xp3" type="rect" doc="coeur.gif"
        from="space (3d) (0 81 745) (30 290 238)" />
<xptr id="xp4" type="line" doc="coeur.gif"
        from="space (3d) (0 44 751) (30 326 269)" />
</ptrGrp>
```

Chaque balise *<xptr>* est ensuite associée au moyen de la balise *<corresp>* à un élément XML/TEI (par l'intermédiaire de leur identifiant respectif), comme illustré ci-après.

```
<correspGrp>
<corresp targets="l1 xp1" />
<corresp targets="l2 xp2" />
<corresp targets="p1 xp3" />
<corresp targets="l4 xp4" />
</correspGrp>
```

Rappelons ici que ce code XML/TEI est généré automatiquement par notre outil d'édition au moment de la création interactive de la transcription. A l'écran l'utilisateur visualise une interprétation graphique de ce code (Figure 2, partie c). Comme dans un navigateur classique (par exemple Netscape Navigator), on applique des fonctions de rendu graphique au contenu des balises XML.

Correspondance Balises XML / Objets Graphiques

Dans notre application, les aspects texte et image sont fortement couplés car ils sont intégrés dans une structure arborescente interne unique permettant des interactions complexes entre ces deux types de ressources.

Pour ce faire, l'architecture de notre système repose sur une nouvelle boîte à outils graphique. Ce nouveau toolkit, appelé *Ubit* [10,11], est essentiellement composé d'objets génériques et modulaires, appelés *briques de base*, qu'il est possible de combiner librement afin d'obtenir une grande variété de composants graphiques spécialisés. Cette conception offre plusieurs avantages. L'utilisation des briques de base permet de créer des objets graphiques qui correspondent aux besoins du domaine d'application. En effet, la généricité et la simplicité de ces briques les rendent facile à dériver en une grande variété d'objets pouvant accomplir des tâches spécifiques. Une brique peut par exemple implémenter un objet spécifique comme un élément ou un attribut XML. Il est ainsi possible de disposer d'un grand nombre d'objets graphiques sans pour autant sacrifier les performances. Par opposition, la plupart des toolkits graphiques fournissent des composants qui implémentent de nombreuses caractéristiques standards (même lorsqu'ils sont utilisés pour des tâches spécifiques) et, en conséquence ont tendance à utiliser de grandes quantités de mémoire. Ce schéma de conception impose de sérieuses limitations sur le nombre d'objets graphiques qui peut raisonnablement être utilisé. En conséquence, les logiciels efficaces tentent généralement de mixer différents styles de programmation et d'introduire des astuces de codage qui accroissent leur complexité tout en limitant leurs capacités.

Au niveau de notre application, nous tirons parti des spécificités du toolkit Ubit de la façon suivante. Au lieu de manipuler des représentations internes séparées de l'arbre XML et de l'arbre d'instanciation des objets graphiques, on les rassemble dans un graphe unique. Chaque nœud de l'arbre XML représentant le document est une brique qui associe les caractéristiques XML et leurs fonctions de rendu graphique. Cette représentation facilite la construction d'éditeurs interactifs puisque chaque objet (les balises) apparaissant effectivement à l'écran correspond à un objet réel et unique de la représentation interne. Ainsi, la représentation textuelle affichée dans notre système (Figure 2, partie c) peut être vue comme une projection de l'arbre XML/Ubit qui est automatiquement gérée (et mise à jour à l'écran) par le toolkit.

Ce schéma de conception possède également l'avantage de fournir directement la correspondance inverse de la représentation visuelle vers les données internes. La correspondance entre la position du curseur sur le texte et les nœuds XML est connue en permanence de telle sorte qu'un déplacement de la souris sur l'écran est équivalent à un déplacement de pointeur dans l'arbre XML/Ubit.

Enfin, nos nœuds XML/Ubit contiennent également les données spatiales qui permettent de lier le contenu des images avec leurs parties textuelles (dans la transcription), c'est-à-dire les informations de couplage texte/image. Les nœuds XML/Ubit intègrent ainsi trois types de données dans une représentation unique :

- les caractéristiques des éléments et des attributs XML (et leur contenu),
- les caractéristiques graphiques du rendu approprié au texte structuré dans la transcription et la vue XML,
- les coordonnées spatiales de ce texte dans les images de manuscrits.

Un autre point intéressant correspond au fait que les caractéristiques graphiques standards pour le rendu du texte ne sont pas stockées dans les nœuds XML/Ubit mais dans des objets de *style* séparés. Ces styles, une autre caractéristique standard du toolkit, fournissent un moyen simple et efficace de paramétrer les briques Ubit. Encore une fois, cette caractéristique facilite la translation entre les “ mondes ” XML/Ubit (document structuré / interface graphique), avec un double niveau de correspondance :

- les balises et les attributs XML / les briques Ubit pour le codage,
- les feuilles de style XML / les objets de style Ubit pour le rendu.

EXPLORATION D'UN ESPACE HYPERMEDIA

Dans la partie précédente nous avons abordé la problématique de l'accès aux documents par leur contenu. La technique d'annotation interactive proposée est utilisée pour associer du texte à des parties d'une image afin de pouvoir les indexer. Dans le cas de notre exemple d'images de textes manuscrits (Figure 1), cette méthode propose une solution à l'accès d'images difficilement exploitable par traitement automatique.

Dans cette deuxième partie on s'intéresse à des méthodes plus intuitives d'accès aux données, souvent associées à des systèmes de type hypermédia. Ces systèmes organisent l'information en termes de nœuds et de liens. L'exploration s'effectue simplement par associations d'idées, en suivant les liens rencontrés. Nous décrivons ci-dessous des techniques d'interaction et de visualisation que l'on a développées pour aider à trouver et à retrouver des documents dans un espace documentaire hypermédia. Le travail réalisé a été appliqué, comme exemple pratique, à notre espace de données littéraires. Cette base documentaire se compose d'images de textes manuscrits, des documents XML/TEI correspondants (créés par transcription) et de documents connexes. Une organisation de type hypermédia, basée sur des liens codés en XML, permet d'interconnecter tous ces éléments et fournit la base du système de navigation.

Navigation

Pour réduire les problèmes de désorientation [12] généralement associés à la navigation hypermédia, nous proposons un système, appelé *ZoomTree*, dédié à l'exploration de documents en contexte. Ce système, introduit dans [14,15], aide l'utilisateur à localiser plus facilement un document particulier et à comprendre ses relations avec les documents qui lui sont associés via des hyperliens. Nous nous contentons ici d'en rappeler les grandes lignes en insistant sur l'interaction proposée pour manipuler les documents.

Notre système permet d'extraire du réseau d'hyperliens des petits groupes de documents et de visualiser simultanément, à un niveau de détail variable, leur contenu. Chaque groupe est composé de quelques documents connectés entre eux par des hyperliens. Chaque groupe correspond à une sous-partie de l'hyperespace exploré: c'est un *aperçu local*. Ces vues locales se matérialisent à l'écran sous forme de hiérarchies de fenêtres (Figure 5). Les documents sont quant à eux affichés dans ces fenêtres que l'utilisateur peut déplacer ou redimensionner par manipulation directe au moyen de la souris. Les liens hiérarchiques (ceux affichés dans la représentation) qui partent d'une fenêtre correspondent à des hyperliens qui partent du document contenu dans la fenêtre.

Figure 5: Les aperçus, des vues locales sur des sous-ensembles de l'espace documentaire.

Dans le même cadre d'idée, on peut citer PadPrints [7]. Ce système construit dynamiquement une hiérarchie qui représente l'historique d'une navigation Web. Chaque page Web est représentée par une image. L'utilisateur garde une représentation visuelle des pages consultées. Ceci aide l'utilisateur à savoir d'où il vient et quel chemin il a suivit. Les retours en arrière peuvent s'effectuer simplement. Cependant, l'utilisateur ne dispose pas d'information l'aidant à effectuer ses choix de navigation. De même il semble difficile de comparer des documents distants en termes de liens.

Plusieurs avantages se dégagent de notre représentation. Tout d'abord, l'accès au contenu des documents s'effectue en contexte. Comme illustré à la Figure 5, plusieurs documents peuvent être visualisés en détail (des points de focus) alors que d'autres, formant leur contexte, sont affichés à taille réduite. L'utilisateur dispose ainsi d'une représentation qui l'aide à comprendre la place d'un document particulier dans un ensemble plus important. Ceci facilite l'interprétation du document consulté, celui-ci étant rattaché à des données qui lui sont associées. Ensuite, de par sa conception, notre système permet de définir des représentations multi-focus. Ceci nous semble pouvoir favoriser les associations d'idées en permettant la comparaison entre différents documents. Enfin, proposer à l'utilisateur une représentation visuelle des documents (mise en page, nombre et forme de leurs composants, etc.) accessibles par sélection d'un hyperlien, par rapport au document courant, peut faciliter les choix de navigation.

Interaction. Un point important du système concerne l'interaction, basée sur un paradigme de *zoom direct*. Ce type d'interaction est utilisé à la fois pour agir sur la représentation du contenu des documents et pour organiser l'espace d'affichage (gestion du focus et du contexte, manipulation de groupes de documents, etc.). Dans ZoomTree le zoom ne sert pas à se déplacer dans l'espace documentaire, comme par exemple dans le cas du système Pad++[2]. Dans ce dernier, l'accès aux données s'effectue par une succession de zoom. Lorsqu'un zoom est appliqué, de nouveaux objets apparaissent et le détail de ceux déjà présents augmente. Les objets de grande taille passent derrière le point de vue de l'utilisateur et disparaissent. Il s'avère cependant difficile de comprendre la structure de l'espace exploré et de trouver un document particulier. De même la comparaison entre objets distants ne peut pas s'effectuer simplement (ce qui ne favorise pas les associations d'idées). Enfin le zoom est global et non borné. Il s'applique à l'ensemble de l'espace documentaire, ce qui engendre de lourdes et fréquentes opérations de mise à jour.

Dans ZoomTree, le zoom s'applique aux objets de la représentation: il est local. Changer interactivement la taille d'une fenêtre induit une opération de zoom sur le document qu'elle contient. Tous les éléments du document (texte, images, etc.) sont redimensionnés tout en respectant la mise en forme initiale : ils sont toujours disposés de la même manière les uns par rapport aux autres. Ce comportement diffère de ceux rencontrés par exemple dans les navigateurs Web. En effet, changer la taille de la fenêtre d'un navigateur courant (comme Netscape Navigator) entraîne une réorganisation spatiale des éléments contenus dans la page Web visualisée. En comparaison, notre choix de conception offre plusieurs avantages. Les utilisateurs ne sont pas perturbés par des changements fréquents de mise en forme dus aux changements de taille. Les documents affichés à petite taille restent ainsi visuellement identifiables, leur mise en page étant préservée.

L'interaction proposée unifie donc deux types d'opérations en une action : le redimensionnement des fenêtres (gestion et organisation de l'espace d'affichage) et le zoom de leur contenu. Les zooms interactifs permettent également de retailler/manipuler des ensembles de documents (les fenêtres correspondantes). Il est ainsi possible de redimensionner un groupe de fenêtres en une seule interaction. Cette possibilité évite à l'utilisateur d'avoir à définir la répartition entre focus et contexte en désignant explicitement la taille de chaque fenêtre, ce qui pourrait s'avérer fastidieux. C'est également un moyen de simplifier les opérations d'organisation spatiale des fenêtres. Plusieurs techniques de zooms " collectifs " (de groupes) sont ainsi proposées à l'utilisateur [15]. Dans la même optique, et ce afin de préserver la cohérence du paradigme d'interaction proposé, le zoom est également identifié à l'opération de fermeture. Lorsqu'une fenêtre approche d'une taille minimum, elle est automatiquement fermée.

Organisation et Repérage

Dans un espace hypermédia il est souvent difficile de retrouver rapidement des documents spécifiques, ou encore de rattacher un document à un contexte personnel d'utilisation. Une solution consiste à sauvegarder quelque part une représentation des documents en vue d'une utilisation future.

Dans un environnement WIMP classique, l'utilisateur peut positionner des icônes sur le bureau de travail afin de retrouver plus facilement les documents auxquels elles sont associées. Le nombre d'icônes affichées est cependant limité. De plus, elles ne donnent pas d'information sur le contenu des documents. Le système Treemap [8], utilisé pour afficher des vues d'ensemble sur des collections de documents, propose une autre forme d'organisation spatiale 2D. Ce qu'il est important de remarquer c'est que la disposition spatiale des éléments affichés est générée automatiquement. Ceci rend difficile l'interprétation de la représentation. Par opposition, des environnements 3D comme Workscape [1] ou le Web Forager [3] autorisent l'utilisateur à positionner lui-même les documents afin de l'aider à les retrouver ultérieurement. L'utilisation de la 3D complique toutefois l'interaction. Le système Data Mountain [16] permet de placer interactivement des images réduites de pages Web sur un plan incliné. C'est la mémoire spatiale et visuelle de l'utilisateur qui l'aide à retrouver les documents. L'effet de perspective permet de gérer au mieux l'espace d'affichage et d'augmenter ainsi le nombre de documents qu'il est possible d'afficher. Cependant ce nombre reste limité. Plus important, il ne propose pas de mécanisme pour repérer des groupes de documents, ni pour accéder seulement à certains types de documents.

Notre système permet la gestion d'une hiérarchie de surface de visualisation 2D. Sur chacune de ces surfaces, appelées *tables de travail*, l'utilisateur peut stocker une représentation iconique des documents de l'espace hypermédia, comme illustré à la Figure 6. Chaque table de travail peut être vue comme un espace de bookmark visuel.

Chaque icône est une image créée au moment de la consultation du document correspondant (par capture d'écran). Une fois l'icône créée, l'utilisateur la positionne sur la surface de visualisation à l'endroit et à la taille désirée. Chaque icône peut être déplacée interactivement par " drag and drop ". Leur taille peut également être modifiée par manipulation directe en " tirant dessus " au moyen de la souris, ce qui entraîne un zoom de l'image. Ceci permet de mettre en correspondance le niveau de détail d'une icône et l'intérêt porté au document correspondant à un moment donné, un objet plus grand attirant plus facilement l'attention. De plus, l'utilisateur peut regrouper, en terme de localisation spatiale, des documents en fonction de son propre modèle cognitif des données.

Figure 6: Organisation logique et personnalisée des documents. Partie A: cette icône repère un groupe de documents particulier (un dossier de la hiérarchie).

L'utilisateur définit ainsi interactivement sa propre organisation conceptuelle de l'espace hypermédia, celle-ci pouvant évoluer selon son activité. Il établit des schémas de classification et de relation entre les documents en se basant sur des propriétés visuelles et spatiales. A chaque exploration, l'utilisateur se retrouve dans un environnement familier. Ceci lui évite d'avoir à reconstruire un modèle mental de l'hyperespace : où se trouvent les documents intéressants ? comment faire pour y accéder ? etc., réduisant du même coup la surcharge cognitive associée à ces différentes tâches.

Pour soutenir notre approche on peut faire référence à une autre étude [4] qui a déjà montré, dans le cadre de bookmarks Web, que la mémoire spatiale de l'utilisateur (plus le fait qu'il organise les objets lui-même " à la main ") et l'utilisation de représentations iconiques permettent de retrouver plus rapidement des documents préalablement stockés. Cette étude montre aussi que ces aides permettent de se remémorer plus facilement les informations acquises sur un espace informationnel après une longue absence d'utilisation.

Afin de ne pas être limité à une seule table de travail (contrairement au Data Mountain), notre environnement permet la création et la gestion d'une hiérarchie de telles surfaces. La conception du système se base sur le modèle classique d'un explorateur de fichiers, comme illustré à la figure 7. Chaque élément de la hiérarchie (un dossier) correspond à une table de travail. Une telle surface peut être dédiée à une tâche spécifique, à un sujet donné ou encore à un espace hypermédia particulier. Le système propose donc un double niveau d'organisation: le premier au niveau de la hiérarchie, le second au niveau d'une table de travail, les documents pouvant être regroupés par localisation spatiale. En plus des documents, on affiche sur chaque table de travail des icônes pour représenter les dossiers qu'elle contient. Les images associées à ces icônes permettent d'indexer simplement des groupes de documents. Dans l'exemple de la Figure 6, l'icône A correspond à un dossier où sont stockées des images de manuscrits contenant des ratures.

Figure 7: Résultat d'une transformation Fisheye. L'arborescence des tables de travail est affichée en haut à gauche.

Enfin, une transformation Fisheye [6] permet de focaliser sur une zone d'intérêt en une seule interaction. Une opération suffit à changer la taille et la position des documents affichés en fonction de l'intérêt de l'utilisateur. Ceci lui évite d'avoir à effectuer de trop nombreuses interactions. L'utilisateur clique à un endroit de la surface d'affichage. Le point désigné devient le centre de la vue Fisheye. Les éléments proches de ce point sont agrandis alors que ceux plus éloignés voient leur taille diminuée. La répartition de l'espace alloué à chaque élément est modifiée, chacun d'eux reste cependant affiché sur la surface. La figure 7 montre le résultat d'une telle opération appliquée à la Figure 6 (la zone de focus se trouve en bas à droite).

Nous avons également appliqué notre système au stockage de pages Web, fournissant ainsi une alternative au mécanisme classique de bookmarks proposé par les navigateurs actuels (Netscape Navigator, Internet Explorer).

CONCLUSION

Nous avons voulu montrer que les espaces hypermédia, et plus généralement tout type d'espace informationnel, peuvent tirer avantage de l'intégration de techniques d'IHM et de visualisation d'information pour faciliter l'accès aux documents.

Le travail présenté a été effectué dans le cas d'un espace hypermédia composé en grande partie d'images de textes manuscrits littéraires. Ce travail a été réalisé avec l'aide d'équipes de spécialistes d'œuvres littéraires anciennes et modernes, en particulier des chercheurs de l'ITEM et de l'IRHT. Deux approches complémentaires sont proposées pour l'accès aux documents. On montre tout d'abord que des techniques interactives d'annotation permettent d'associer des parties d'une image bitmap à du texte structuré codé en XML. Des traitements automatiques étendus peuvent alors être effectués sur les balises et leur contenu. Il est ainsi possible, entre autres, d'accéder aux images par indexation. Du point de vue de l'implémentation, nous proposons l'extension d'un mécanisme de pointage XML permettant de définir des liens entre des éléments d'un document XML et des zones d'une image bitmap. Plus généralement, ces techniques d'annotation peuvent être appliquées à tous types de documents numérisés stockés sous forme d'images.

La seconde approche consiste à accéder aux documents par exploration interactive de l'espace informationnel. L'environnement développé contient un système de navigation pour aider à trouver les documents et un système de gestion pour aider à les retrouver. Le système de navigation permet de visualiser le contenu des documents en contexte et favorise les associations d'idées. Le système de gestion permet à l'utilisateur de construire interactivement sa propre organisation logique sur les données consultées afin de faciliter leur réutilisation.

Une des prochaines étapes consistera à améliorer les possibilités d'annotation hypertextuelle de notre environnement. Cet aspect de notre travail est à rapprocher de la notion de lecture active. Ce concept, qui vise à atténuer la traditionnelle séparation entre phases d'édition et de lecture en permettant un enrichissement continu des documents par le (ou les) lecteur(s) est d'autre part probablement voué à des développements intéressants du fait de l'arrivée de nouveaux matériels appropriés (du type livre ou papier électronique [13]). Une autre possibilité intéressante consisterait à coupler un moteur de recherche à nos systèmes de visualisation. Le résultat d'une recherche sur un ensemble de fichiers XML pourrait alors être représenté visuellement. Enfin, il faut mettre en œuvre des procédures d'expérimentation afin d'évaluer la validité des systèmes proposés et leurs apports.

BIBLIOGRAPHIE

1. Ballay J.M. *Designing Workscape: An Interdisciplinary Experience*. ACM Conference on Human Computer Interaction (CHI'94), pp. 10-15.
2. Berderson B., Hollan J.D., Perlin K., Meyer J., Bacon D. and Furnas G. *Pad++ :a Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics*. Journal of Visual Languages and Computing, 7:3-31, 1996.
3. Card S.K., Robertson G. and York W. *The WebBook and the WebForager : An Information Workspace for the World Wide Web*. ACM Conference on Computer Human Interaction (CHI'96), pp. 111-117.
4. Czerwinski M.P., van Dantzich M., Robertson G. and Hoffman H. *The Contribution of Thumbnail Image, Mouse-Over Text and Spatial Location Memory to Web Page Retrieval in 3D*. IFIP Conference on Human-Computer Interaction (INTERACT'99), pp. 163-170.
5. Editors of the William Blake Archive. *The Persistence of Vision: Images and Imaging at the William Blake Archive*. RLG DigiNews4.1.
6. Furnas G. *Generalized Fisheye Views*. ACM Conference on Computer Human Interaction (CHI'86), pp. 16-23.
7. Hightower R.R., Ring L.T., Helfman J.I., Bederson B.B. and Hollan J.D. *Graphical Multiscale Web Histories : a Study of PadPrints*. ACM Conference on Hypertext and Hypermedia (HT'98), pp. 58-65.
8. Johnson B. and Shneiderman B. *Space-filling Approach to the Visualization of Hierarchical Information Structures*. IEEE Visualization'91, pp. 284-291.
9. Lecolinet E., Role F., Robert L., Likforman-Sulem L. and Lebrave J-L. *An Integrated Reading and Editing Environment for Scholarly Research on Literary Works and their Handwritten Sources*. ACM Conference on Digital Libraries (DL'98), pp. 144-151.
10. Lecolinet E. *A Brick Construction Game Model for Creating Graphical User Interfaces: the Ubit Toolkit*. IFIP Conference on Human-Computer Interaction (INTERACT'99), pp. 510-518.
11. Lecolinet E. *Réification et réplication dans les interfaces graphiques : le toolkit Ubit*. Onzièmes Journées Francophones sur l'Interaction Homme Machine (IHM'99). Page Web Ubit: http://www.enst.fr/~elc/ubit.
12. Nielsen J. *Hypertext and Hypermedia*. Academic Press, 1990.
13. Price M.N., Golovchinsky G., Schilit B.N. *Linking by Inking : Trailbalzing in a Paper-like Hypertext*. ACM Conference on Hypertext and Hypermedia (HT'98), pp. 293-294.
14. Robert L. and Lecolinet E. *Browsing Hyperdocuments with Multiple Focus+Context Views*. ACM Conference on Hypertext and Hypermedia (HT'98), 1998.
15. Robert L. and Lecolinet E. *Couplage de représentations globales et locales pour l'exploration d'hyperdocuments*. Dixièmes Journées Francophones sur l'Interaction Homme-Machine (IHM'98), pp. 27-34.
16. Robertson G., Czerwinski M., Larson K., Robbins D.C., Thiel D. and Van Dantzich M. *Data Mountain : Using Spatial Memory for Document Management*. ACM UIST'98, pp. 153-162.
17. Sperberg-McQueen C.M., Burnard L. *Guidelines for electronic text encoding and interchange (TEI P3)*. ACH-ACL-ALLC Text Encoding Initiative, 1994.
18. World Wide Web Consortium, *Extensible Markup Language (XML) 1.0*, W3C Recommendation. http://www.w3.org/TR/1998/REC-xml-19980210.