Réflexions sur la modélisation des documents

Veronika Lux-Pogodalla, Jean-Yves Vion-Dury

Réflexions sur la modélisation des documents

Veronika Lux-Pogodalla*‡ et Jean-Yves Vion-Dury†‡

***ATILF CNRS** (sabbatique),
44 avenue de la Libération, B.P. 30687, 54063 Nancy Cedex
vlux@atilf.fr
† **INRIA** (sabbatique),
655 avenue de l'Europe, 38334 Montbonnot
Jean-Yves.Vion-Dury@inrialpes.fr
et
‡**Xerox Research Center Europe** 6, chemin de Maupertuis,
38 240 Meylan

Résumé

A l'heure du numérique, plus que jamais, nous avons besoin d'abstractions et de modèles susceptibles d' inspirer des approches novatrices pour la création et le traitement du document, le raisonnement sur des transformations complexes de documents et la caractérisation de leurs propriétés. Nous cherchons ici à mieux saisir les documents d'aujourd'hui et à mener une réflexion sur les documents de demain, en nous appuyant largement sur l'analyse des évolutions technologiques au cœur desquelles il se situe.

***Mots-clés** : modèle de document, transformations de documents*

Abstract

Today, we more than ever need useful abstractions in order to reason on complex document transformations, to assert properties on document manipulation systems, and to inspire perhaps revolutionary approaches of document creation and processing. This paper sketches perspectives and proposes markers toward inventing the next document generation.

Keywords :document model, document transformation

1. INTRODUCTION

Nous cherchons ici à mieux saisir le document d'aujourd'hui, en nous appuyant largement sur l'analyse des évolutions technologiques au cœur desquelles il se situe. Notre approche concerne en priorité de ce que [34] qualifie de « document en tant que forme », quoique l'examen de l'évolution des traitements documentaires nous fasse probablement sortir de ce cadre.

Notre objectif à long terme est de proposer un modèle documentaire[1] souple, évolutif et calculable. Dans la présente contribution, nous livrons quelques réflexions préliminaires et nous nous appliquons à préciser certaines propriétés qu'un tel modèle devrait vérifier. Dans la première section, un *modèle conceptuel* simple est proposé, qui met en évidence les liens entre traitements et niveaux de représentation des documents. Nous nous concentrons ensuite sur les documents en tant qu'objets calculables, et examinons quelques *modèles opérationnels*, c'est-à-dire supports de réalisations informatiques, pour mettre en évidence quelques aspects essentiels du traitement documentaire. Dans la deuxième section, nous analysons en détails quelques-unes des opérations qui ont pour objet le document et qui se font désormais avec la médiation d'ordinateurs. Cette réponse partielle et personnelle à la question « Que faisons-nous avec les documents ? » doit contribuer aussi à mieux cerner les documents d'aujourd'hui

En guise de conclusion, nous proposons quelques desiderata sur les modèles de documents, qui sont une contribution pour de futurs débats sur une « science du document » dont les fondations restent à définir. D'une part, comme le souligne [34], les documents intéressent de nombreuses disciplines différentes (par exemple : linguistique, sémiologie, informatique, théorie de l'information) entre lesquelles une véritable collaboration sera requise. Par ailleurs, les documents sont au cœur de technologies en rapide évolution dont il est nécessaire mais difficile de savoir s'abstraire.

---

[1] Le mot « documentaire » est utilisé dans cet article avec le sens particulier de « lié aux documents »..Il faut donc lire ici « modèle de document ».et « traitement du document »,. De même dans la suite.

2 DOCUMENTS ET MODELES DE DOCUMENTS

2.2 Modèles conceptuels de documents

Les systèmes complexes sont souvent abordés selon différentes dimensions dans lesquelles les objets d'étude peuvent être projetés, c'est-à-dire réduits en des formes simplifiées, sans perdre le sens relatif à leur étude [38].

Afin de faciliter l'analyse des différents usages des documents, nous proposons dans cette section un modèle conceptuel pouvant être considéré comme une première étape vers un modèle formel. Il convient selon nous de distinguer les modèles conceptuels des modèles opérationnels : les premiers aident à comprendre, expliquer et analyser les problèmes tandis que les seconds offrent des moyens pratiques de résoudre les problèmes d'ingénierie ou d'implanter des architectures efficaces pour le traitement logiciel. Ainsi, le modèle conceptuel discuté dans cette section est-il basé sur une notion, certes abstraite, d'espace documentaire à dimensions multiples, mais il apporte un éclairage stimulant sur les propriétés du document ainsi que sur les transformations qu'il est susceptible de subir.

Idéalement, chaque dimension devrait être « orthogonale »aux autres dimensions, en ce sens qu'une modification d'une de ses grandeurs ne devrait pas affecter les autres dimensions. Par exemple, une variation de la localisation spatiale d'un document (transport) ne doit pas entraîner de modification de son contenu.

Nous avons identifié l'espace et le temps comme des dimensions physiques (indissociables du processus de restitution), contenu et forme comme des dimensions logiques. De plus, il nous est apparu ultérieurement nécessaire de distinguer les dimensions intrinsèques et extrinsèques aux documents. Ainsi, l'espace intrinsèque permet de décrire l'organisation interne de la présentation d'un document (comme les positions relatives des figures et paragraphes dans une page), alors que l'espace extrinsèque englobe les opérations d'échange ou de diffusion de documents. Le temps intrinsèque quant à lui est utile pour la modélisation de flux vidéo synchronisés, tels que mis en œuvre dans les documents multimédia, alors que le temps extrinsèque est adapté, par exemple, à la modélisation des diverses phases de création de versions au sein de dossiers documentaires. Par ailleurs, si le temps est par nature

scalaire[2], il peut être continu ou discret selon les besoins spécifiques de la modélisation.

Chaque dimension possède ses propres propriétés et règles d'organisation. Examinons par exemple la dimension forme.

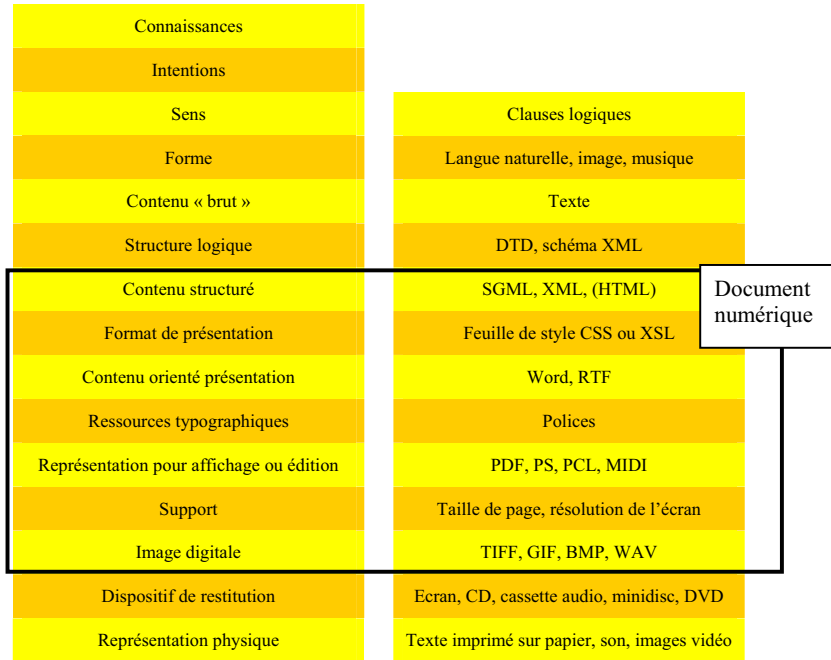

FIG. 1: ***Vers une abstraction croissante vis-à-vis du médium physique***

La figure 1 organise diverses formes documentaires de bas en haut, selon une abstraction croissante vis-à-vis du médium physique. Plus la forme est abstraite, plus difficiles sont les processus de transformation visant au formatage sur support physique, qu'il soit papier, écran ou

[2] Il se prête aux comparaisons d'ordre et aux opérations proportionnelles.

système de restitution audio : les transformations applicables aux documents sont donc hautement conditionnées par leur forme. Les techniques de compression, par exemple, varient notablement selon le niveau d'abstraction : des techniques d'algorithmes sans pertes sont obligatoires pour des documents sous forme logique (comme les pages HTML), mais cette contrainte peut être relâchée pour des formes de plus bas niveau, telles que des documents codés sous format image (comme TIFF).

Afin d'illustrer cette notion, nous donnons à présent quelques exemples de formes documentaires d'abstraction croissante :

- images digitales, ou son numérisé (TIFF, GIF, WAV, MP3) : les unités élémentaires sont des pixels ou des échantillons sonores. Les transformations associées appartiennent au domaine du traitement du signal (corrections colorimétriques ou physiologiques, interpolation de pixels et autre filtres de convolution[3], etc.). Le stockage du document est coûteux, et fonction de la qualité du rendu, ce qui justifie l'utilisation d'algorithmes de compression de données. Notons que la modification du contenu est pratiquement impossible à ce niveau.
- description de pages (Postscript, PDF) : les entités élémentaires sont des caractères, graphiques, images associés à des informations de positionnement dans la page. Le stockage est raisonnablement compact, le document n'est que très partiellement modifiable, et ce dans la limite d'une page. Le rendu graphique requiert un traitement pouvant être fort complexe.
- contenu orienté présentation (MS-word, RTF) : les entités élémentaires sont plutôt hétérogènes (informations de style, graphiques, indications de structure logique, etc.). Le stockage est relativement compact, le document est modifiable par édition, mais peu de transformations sont possibles, de par les faibles propriétés de sa forme. La restitution du document requiert des pilotes logiciels spécialisés.
- document structuré (SGML, XML) : les entités élémentaires sont des éléments structuraux (ou logiques) dont la signification est paramétrable. Les règles d'organisation logique sont explicites et modifiables séparément des instances documentaires (schémas documentaires). La validité structurelle d'un document par rapport à un schéma peut être vérifiée via des algorithmes génériques. Le document est modifiable, mais des outils d'édition spécialisés pour un

[3] Technique générique de filtrage numérique basée sur la multiplication du signal par une matrice de coefficients réels, appelée « noyau de convolution ».

schéma particulier peuvent être requis dans le cas de schémas complexes ou pour des utilisateurs non spécialistes.

La figure suivante (Fig. 2) illustre trois transformations réalisées à trois niveaux de forme différents : la photocopie optique, au niveau de la représentation physique ; la photocopie numérique, au niveau de la représentation numérique et une hypothétique transformation effectuée par un photocopieur traducteur, qui serait nécessairement réalisée à un niveau d'abstraction beaucoup plus haut,: celui du sens.

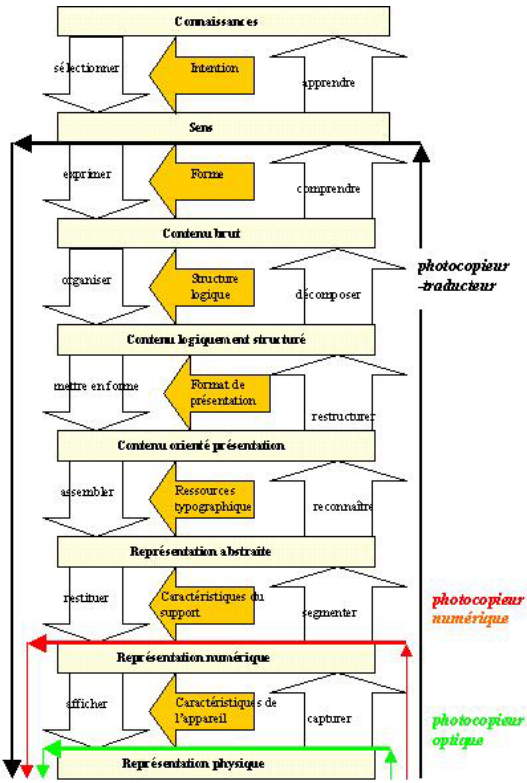

FIG. 2 : *trois photocopies différentes*

Les transformations sont représentées sur la figure 2 par des chemins (flèches) traversant les différentes couches d'abstraction. Plus ces chemins sont longs, plus la transformation est complexe du point de vue quantitatif. Cette métaphore reflète bien le fait que l'accroissement de

complexité quantitative (c.-à-d. en terme du nombre d'opérations à accomplir) est directement liée aux longueurs des chemins. Par contre, la complexité qualitative des transformations est mal reflétée par la figure dont la symétrie est une illusion : en effet, la partie « ascendante » d'une transformation qui construit une abstraction est bien plus complexe que la contre-partie « descendante » – qui réduit l'abstraction. Il suffit pour s'en convaincre de comparer la complexité des procédés d'OCR (Optical Character Recognition) aux procédés de restitution tels que mis en œuvre dans l'impression.

Cette limite de notre modèle purement conceptuel motive l'examen dans la prochaine section de modèles opérationnels capables de capturer et refléter cette complexité.

2.2 Modèles documentaires opérationnels

Les modèles opérationnels sont par nature intimement liés aux algorithmes de transformation documentaire. Ce que l'on appelle habituellement « format » cache *de facto* une notion très concrète de modèle de document : des hypothèses fortes directement liées aux modalités de traitement sous-tendent ces formats de données qui sont avant tout conçus pour le stockage et les échanges de documents.

Ainsi, le format PDF d'Adobe [46], qui est construit autour d'un modèle général à base de pages, se focalise d'avantage sur la publication via un réseau électronique que son prédécesseur Postscript [47], qui lui était résolument orienté vers l'impression papier. En conséquence, le premier incorpore de puissants mécanismes de compression de données paramétrables, susceptibles d'être finement adaptés à la nature des contenus documentaires. Par ailleurs, la dépendance aux ressources typographiques, potentiellement très complexes (des milliers de fontes différentes sont utilisables) et source de nombreux problèmes est maîtrisée par des mécanismes d'inclusion directe. Si les deux formats reposent sur un modèle graphique très riche, seul le premier prend en compte la notion de transport électronique. Notons également que Postscript conçoit un document comme le résultat d'un calcul (les instructions graphiques qui dessinent les pages – postscript est un langage de programmation !) alors que PDF voit les documents comme des descriptions déclaratives de pages et, à ce titre, s'abstrait de la complexité calculatoire. Ainsi, le dernier né des formats Adobe reflète clairement une évolution vers une abstraction accrue, vers plus d'indépendance vis-à-vis du medium physique et de ses infrastructures associées.

Une classification rapide de quelques modèles parmi les plus connus apporte un éclairage supplémentaire sur l'interdépendance format-traitement et confirme la tendance vers toujours plus d'abstraction :

- TeX, Metafont [20,21] est focalisé sur la typographie, offrant de très grandes possibilités pour la création de fontes et la maîtrise du processus de formattage. Des algorithmes dédiés sont mis en œuvre (de manière plutôt opaque à l'utilisateur), et le document est conçu comme le résultat d'un processus de compilation. Le contenu documentaire est mélangé aux instructions de programmation, qui reflètent un modèle de présentation riche et complexe. L'utilisateur doit en devenir un expert afin de pouvoir tirer parti de toute sa puissance.
- les modèles Wysiwyg (*What You See Is What You Get*), tels que mis en œuvre par Microsoft Word, privilégient le processus d'édition. Ils sont peu abstraits et focalisés sur la présentation. Les opérateurs qui la régissent sont directement montrés et manipulés par l'utilisateur via des icônes et des décorations graphiques animées. Bien que difficilement séparable du contenu documentaire en général, la présentation est donc contrôlable de manière intuitive. Le document ne demande pas de phase de compilation pour être visualisé, mais est mis à jour incrémentalement. Comparés aux modèles orientés typographie, les modèles Wysiwyg offrent des rendus de moindre qualité graphique, mais une réelle interactivité et confère au rédacteur une « conscience du document » (*document awareness*). Le document n'est (quasiment) pas modifiable en dehors du processus d'interaction offert par l'éditeur.
- les modèles structurés (Grif [35], XML [71]) sont architecturés sur le principe de la séparation stricte du contenu et de la présentation, qui offre au rédacteur l'avantage de se focaliser sur la structure interne de son document (le choix et l'organisation de son contenu : titre, chapitre, section, etc.), et de se reposer sur des mécanismes externes pour définir et produire la présentation. L'organisation logique d'une famille de documents peut être décrite par un *schéma*. La validation d'un document particulier est le traitement qui établit d'une part que son contenu est bien formé, et d'autre part qu'il est conforme aux règles de structuration spécifiées par son schéma. Il est remarquable que le principe de séparation conduise à une multiplication des traitements documentaires : validation du contenu, transformation de contenu en contenu (enrichissement, simplification, construction d'index, changement de schéma de référence,…), transformation du contenu en présentation (formatage) et, enfin, transformation de présentation en présentation (adaptation, changement de formats).

Grâce aux fortes propriétés de contenu, ce modèle a permis la gestion de très grands systèmes documentaires tels que des manuels de maintenance d'avions de ligne, et leur validation au regard de standards industriels (comme la DTD ATA100 [49]). Les traitements documentaires sont devenus « ouverts » (indépendants des formats et outils propriétaires), et offrent de ce fait une réelle pérennité. Par ailleurs, le principe de séparation contenu/présentation a permis de simplifier la gestion de la cohérence documentaire, puisqu'un contenu unique peut se décliner vers des formats multiples et variés (papier, CD-ROM, intranet, etc.)

- HTML [54] et HyTime [55] sont spécialisés dans la navigation au sein de réseaux de documents connectés par des « hyperliens ». Le traitement sous-jacent est la sélection du lien et sa résolution dans le visualisateur. Cette opération nommée *transclusion* est résolument tournée vers la présentation dynamique et robuste de documents en réseau.
- Postscript [47] est spécialisé dans l'impression de haute qualité
- PDF [46] et DjVu [51] (prononcer « Déjà Vu ») sont spécialisés dans le transport et la publication en ligne de documents à haute valeur ajoutée typographique. Nous avons évoqué plus haut les particularités de PDF et de son rapport avec Postscript. DjVu est dédié à la visualisation de documents anciens, numérisés et compressés au moyen d'algorithmes utilisant très efficacement la théorie des transformations en ondelettes, qui exploitent les particularités graphiques de ce type de documents.

L'analyse des modèles opérationnels montre certes qu'ils sont liés aux métiers d'origine de leurs concepteurs (édition numérique, micro-informatique,…) mais leur dynamique exprime selon nous une évolution vers une abstraction de forme croissante. Plus précisément, le principe de séparation contenu/présentation ouvre un large champ aux transformations documentaires, dont une des caractéristiques est de favoriser les changements de forme. La notion de validité d'un document (c.-à-d. sa conformité à un schéma type) devient centrale en ce sens qu'elle garantit les propriétés calculatoires nécessaires aux algorithmes transformationnels. Ces évolutions marquent le commencement d'une véritable ingénierie du document sur le plan pratique, et déclenche une très forte dynamique industrielle autour d'un nombre croissant de schémas stratégiques (Smil [58], RDF [60], XML schema [69], Xquery [77], XSLT [76], Xpath [73] parmi les plus importants). Mais conjointement, elles transposent la problématique documentaire dans le champs théorique. En effet, de par la rigueur de ses propriétés formelles, le contenu structuré devient objet mathématique, comme le montrent les

nombreuses études portant sur les langages de spécification de schémas [13 ;18 ; 28 ;16], les processus de validation [29] ainsi que les langages formels de transformation et leurs systèmes de type [40 ;63 ;42]. Ainsi, nous assistons à une situation un peu déconcertante où les développements théoriques ne précèdent pas les avancées industrielles, mais où les deux approches doivent s'efforcer de coexister et évoluer ensemble.

3 TROIS ILLUSTRATIONS

Quels sont aujourd'hui nos usages des documents ? Que faisons-nous et que voudrions-nous faire avec les documents ? On peut chercher des éléments de réponse dans les études de quelques ethnographes du monde du travail ([30 ;32], [3] par exemple, pour les activités de lecture). Mais dans cette section, nous livrons une analyse personnelle de différents traitements du document ou opérations effectuées sur les documents, en cherchant à caractériser l'évolution que l'informatique apporte aujourd'hui dans ces traitements. Nous nous autorisons aussi à imaginer quelques traitements du document pour le futur.

3.1. Réutilisation par copier-coller

De tout temps, tout ou parties du document ont été réutilisées mais ces opérations de réutilisation se trouvent aujourd'hui radicalement modifiées en particulier par:

- la grande facilité de leur réalisation : avec les traitements de texte, le copier-coller se fait sans réécriture ni re-saisie ;
- une énorme quantité de contenus réutilisables, facilement accessibles sur le Web d'une part, dans des systèmes de gestions de documents d'autre part ;
- une connaissance mieux formalisée sur la structure du document, désormais réutilisable avec les DTD SGML ou XML

Dans cette section, nous utilisons les différents niveaux d'abstraction présentés en section 2 pour examiner en détail une micro-opération de

réutilisation qui concerne généralement un fragment de document : c'est l'opération familière de copier-coller[4].

Au niveau physique, le copier-coller est réalisé avec des ciseaux et de la colle, en découpant un fragment du document source (qui se trouve ainsi endommagé) et en le collant dans un espace blanc du document cible[5].

Au niveau du contenu mis en forme, le copier-coller est cette opération simple et déjà familière : dans un éditeur Wysiwyg comme Word, nous sélectionnons un fragment de document avec la souris puis le copions à l'endroit désiré d'un simple clic. La complexité cachée de tels copier-coller nous est quelquefois rappelée lorsque nous copions puis collons dans des applications différentes et que, par exemple, certains caractères spéciaux copiés ne sont pas correctement collés ou qu'une mise en forme (comme le gras ou l'italique) que nous aurions voulu conserver disparaît.

Dans une certaine mesure, ces problèmes ont été résolus avec le développement de formats assurant l'interopérabilité [23]. Mais plus fondamentalement, c'est ici la dualité contenu-présentation dissimulée par le « wysiwyg » qui réapparaît : il serait nécessaire de préciser si le copier-coller concerne le contenu seulement ou le contenu et la présentation, de préciser éventuellement que le fragment copié doit voir sa présentation adaptée à l'environnement cible dans lequel il est collé.

Au niveau du contenu structuré (par exemple avec XML), la dualité structure et contenu devient évidente et une opération de copier-coller peut avoir plusieurs sémantiques possibles selon ce qu'il faut copier (par exemple : soit toute la structure XML et tout le contenu, soit une partie de la structure et le contenu, soit seulement le contenu, soit seulement la structure) et comment il convient de le coller (par exemple : avec un résultat en XML bien formé ou avec un résultat valide par rapport à la DTD du document cible ou avec un résultat valide par rapport à une DTD cible modifiée au cours de l'opération).

---

[4] Faute de place, nous laissons de côté d'autres réutilisations concernant le document plus globalement, et qui sont fréquentes pendant la phase de création car, comme l'observe [23] , la production de nouveaux documents implique généralement la réutilisation d'autres documents.

[5] La photocopie est un mode de copier-coller alternatif qui n'endommage pas le document source.

À ce niveau de contenu structuré, l'opération de copier-coller peut requérir des transformations lorsque la structure du fragment copié n'est pas compatible avec la structure du document cible. Ces transformations sont étudiées par [10b][6].

À de plus hauts niveaux d'abstractions, l'opération de copier-coller pourrait impliquer des transformations complexes comme la transformation d'une image en texte (qui peut nécessiter de l'analyse d'image ou être réalisée plus simplement sur la base d'annotations textuelles associées aux images) ou la transformation d'un texte dans une langue en un texte en une autre langue (par exemple par traduction automatique ou sur la base de mémoires de traduction).

Une vision possible pour le copier-coller du futur serait celle d'une opération paramétrisable autorisant l'insertion du fragment copié dans le document cible soit avec le minimum de changements soit avec toutes les adaptations nécessaires.

3.2 Transformations

Avec les formats de documents structurés (comme SGML et XML), le thème des « transformations » de documents est apparu aussi, terme générique pour désigner une famille d'opérations portant sur les documents. La nécessité de transformer les documents nous semble aujourd'hui accentuée par :

- l'existence de très nombreux « formats »de documents (si l'on peut regrouper sous le terme « formats » des langages aussi différents que XML, HTML, RTF, PDF, GIF, TIFF, etc) qui évoluent constamment. Les standards adoptés sont différents selon les pays (ainsi WML et WAP ne sont pas utilisés au Japon). Et la question se pose aussi de la pérennité des documents d'archives.
- la tendance aux documents « doués d'ubiquité » accentuée par le développement de divers supports (tels le téléphone à écran, l'assistant digital personnel, les ordinateurs de poche, les ordinateurs embarqués dans les voitures). Plus que jamais, les transformations sont nécessaires pour assurer l'interopérabilité, les échanges de documents entre systèmes.

---

[6] L'auteur exprime le problème de façon très générale : étant donné un document appartenant à une classe de documents décrits par une grammaire hors-contexte, comment transformer tout ou partie de ce document afin qu'il soit compatible avec une autre classe de documents, défini par une autre grammaire ?

- la volonté de donner une « vue document », calculée au vol, sur de grands ensembles ou bases de données
- la volonté de personnaliser les documents selon le profil de l'utilisateur, et plus généralement, selon le contexte. Des transformations complexes sont requises, comme celles permettant le passage du texte écrit au son, de l'image à la description textuelle, etc. Les opportunités pour intégrer les résultats de recherche en TAL, en traitement d'images, etc. paraissent nombreuses.

Dans beaucoup d'applications, les transformations de documents sont une problématique clé.

À titre d'exemple, examinons la transformation de « transcodage », destinée à adapter les représentations de documents aux multiples types de supports sur lesquels ils sont livrés aux utilisateurs (ordinateur fixe ou mobile, téléphone mobile et autres gadgets miniatures à écran, écrans géants, etc.) et dont chacun a ses exigences techniques quant au protocole d'échange de données ou au format d'affichage (par exemple : XML, WML, HTML, SVG, PDF).

Le « transcodage » (« transcoding») est déjà une réalité (ainsi pour les utilisateurs de Palm Pilot, un service comme AvantGo [50] fournit du contenu contrôlé et des applications). Les transcodeurs aujourd'hui transforment du HTML ou plus rarement XML en WML (Wireless Markup Language), en HDML (Handheld Device Markup Language), en HTML, etc. Le transcodage opéré par des outils comme WebSphere [62] ou FXPal Digestor ([9]) combine plusieurs transformations élémentaires comme :

- transformation de format d'images (par exemple JPEG en GIF) ;
- réduction d'images ou redéfinition des couleurs, conversion d'images en liens vers ces images, conversion d'images en contenu textuel équivalent (par exemple en utilisant la valeur de l'attribut ALT en HTML) ;
- transformation de structure (par exemple tableaux simples transformés en listes, résumé par extraction de la première phrase de chaque bloc de texte et ajout d'un hyperlien vers le texte complet).

Les petits écrans des appareils portables, leurs moyens d'interaction assez pauvres ou très différents de la souris et du clavier des ordinateurs classiques sont des restrictions physiques qui, ajoutées aux limitations de transmission sur le réseau, amènent de nombreux problèmes de transformations seulement partiellement résolus aujourd'hui. Beaucoup de défis restent à relever dans le domaine des transformations du document :

- amélioration de la qualité de transformations élémentaires (par exemple : amélioration de la transformation de résumé en combinant l'utilisation de transformations de structures du documents et l'utilisation de techniques linguistiques pour traiter le contenu),
- spécification d'un format pivot pour les transformations, afin d'éviter de devoir définir des composants de transformation entre chaque paire de formats,
- construction d'une bibliothèques de composants de transformations de base réutilisables et définition d'une sémantique de composition de ces composants,
- définitions de stratégies pour combiner les composants de transformation de base (par exemple selon les caractéristiques du document à transformer et celles de l'appareil cible) ;
- conception de transformations intégrant les contraintes du format cible qui garantiraient par exemple la validité de la sortie par rapport à une DTD ou à un schéma particulier (ce que ne permet absolument pas une transformation XSL-T).

Une vision pour le futur est celle de services de transformations de documents capables de fournir le document adéquat au moment adéquat et sur l'appareil adéquat. Par exemple, imaginons un chercheur assistant à une conférence à Munich.

- À la mi-journée, sur un grand écran dans le hall, un programme de la conférence détaillé et mis à jour pour l'après-midi est affiché. Remarquant qu'une des communications n'est pas celle initialement prévue, il en demande un résumé, s'assure qu'elle est bien identique à une communication à laquelle il a déjà assisté et décide de se dispenser d'une seconde écoute,
- Sur son assistant numérique personnel, il obtient un programme des événements culturels à l'affiche pendant son séjour à Munich et personnalisés selon ses goûts. Il découvre que quelques oeuvres de Kandinsky sont exposées dans un musée proche, ouvert à l'heure du déjeuner et possédant aussi une cafétéria. Il décide de s'y rende pour une visite et un rapide déjeuner,
- Sur des terminaux dans la rue, il peut consulter une carte du quartier où il se trouve, carte qui donne trace de son itinéraire et grâce à laquelle parvient rapidement au musée,
- Sur son *e-book* dans la soirée, la biographie de Kandinsky qu'il est en train de lire se trouve enrichie avec des notes sur les tableaux qu'il vient de voir.

3.3 Navigation

Il existe depuis longtemps des chemins de traverse pour naviguer dans les (collection de) documents papier (catalogues, tables de matières, index, références bibliographiques, etc.) mais les hyperliens ont radicalement modifié la navigation dans les documents : avec un simple clic, l'utilisateur obtient le contenu du document référencé et en quelques clics, il commence à surfer, un sport inconnu dans les bibliothèques à l'époque du papier. Le Web est par excellence le domaine où se pratique cette navigation ; les recherches sur l'hypertextualité ouvrent aussi des perspectives dans des ensembles plus stables et plus limités de documents [41 ; 19].

Des questions fondamentales se posent alors sur la relation entre la structure de surface linéaire du document classiquement suivie par le lecteur, la structure hiérarchisée intra-document (telle qu'elle est explicitée en XML) et la structure de navigation, intra- et inter-documents, plus semblable à un graphe.

L'évolution vers des structures de navigation plus riches et plus complexes paraît naturelle. Dans une certaine mesure, elle est amorcée avec un standard comme Topic Map (norme ISO/IEC 13 250) ou avec des spécifications comme Xlink [65] ou HyTime [55] (norme ISO/IEC 10744:1992) qui définissent des liens plus riches que ceux du HTML actuel. Et le groupe de travail du W3C sur le « web sémantique » contribue à la faire progresser.

Il nous semble pourtant qu'il reste entre autres choses (1) à offrir des vues synthétiques utiles de support de navigation, qui, si l'on pousse la métaphore de la carte un peu plus loin, pourraient être très différentes des index d'hyperliens, (2) à créér des environnements de navigation permettant aux utilisateurs de voir simultanément carte et « paysage » (c.-à-d.. ensemble de documents) et d'utiliser efficacement l'une pour explorer l'autre.

3.4 Conclusion

Copier-coller, transformer, naviguer : ces trois exemples d'usages de documents illustrent les considérations en terme de modèles précédemment énoncées et leur analyse donne une perspective complémentaire sur les changements en cours. Peut-être ces usages évolueront-ils de façon imprévisible, au gré de nouvelles technologies. En tant que chercheurs, on peut cependant trouver dans ces observations des motivations pour revenir à notre réflexion sur les modèles, modèles

permettant de penser, d'analyser et peut-être d'améliorer voire de prédire ces usages du document. Dans la section de conclusion, nous livrons quelques desiderata pour un modèle de document.

4 VERS UN MODELE UNIFIE DU DOCUMENT

Les nombreux modèles opérationnels qui émergent aujourd'hui des activités du consortium W3C sont conceptuellement très hétérogènes, probablement parce qu'il sont essentiellement des réponses techniques à des problèmes bien identifiés. Ce qui est clairement un point fort du point de vue industriel est un point faible du point de vue de la recherche à long terme, où l'on souhaiterait aborder l'ensemble des problèmes dans un cadre commun, si possible unique.

Il est intéressant d'observer aujourd'hui une situation en quelque sorte « inverse à la normale » où de nombreux travaux de recherche sont conduits par l'émergence des modèles basés sur XML. Ces derniers établissent une bonne base technologique agissant comme facteur d'intégration et de stabilité pour les développements futurs. Il est très probable que les prochaines évolutions dans la modélisation des documents reposent encore sur les modèles XML et capitalisent sur les standards et outils actuels.

Nous observons actuellement l'émergence de nombreux travaux théoriques autour de deux fonctions documentaires très importantes : la validation et la transformation. Loin d'être un hasard, ceci reflète le besoin fondamental d'une caractérisation forte et rigoureuse de l'objet documentaire. Les grammaires et automates d'arbres deviennent un moyen standard de formaliser et d'étudier la structure des documents (voir [13]), certains auteurs ont proposé d'étendre la notion d'arbres aux forêts, afin de généraliser les traitements [18 ; 28 ; 16 ; 29]. Les langages de spécification de schémas sont étudiés sur des bases formelles [22 ; 39 ; 11], ainsi que des langages de sélection et de transformation [44 ; 43 ; [63 ; 15]. Un travail important reste à faire pour intégrer les processus de validation aux systèmes de type de ces langages de transformation [40 ; [77 ; 42], et surtout pour définir des opérateurs de composition de transformations qui propagent et conservent les propriétés logiques des documents. La modularisation et la composition des transformations documentaires sont des facteurs décisifs pour le développement d'une ingénierie documentaire économiquement viable et fiable.

L'intérêt d'une « science du document » semble évident, et celle-ci pourrait s'appuyer sur des langages de modélisation construits sur des bases théoriques fortes, suffisamment puissants pour décrire et caractériser les propriétés des architectures de gestion documentaire complexes (voir les remarques générales de [45] sur les développements formels).

La difficulté principale de cette perspective optimiste serait probablement d'établir le lien adéquat entre les théories des structures de données et du calcul transformationnel d'une part, et une notion spécifique de la perception et de la cognition qui soit applicable dans le domaine documentaire d'autre part.

5 REMERCIEMENTS

Nous remercions en particulier Chris Thompson (principal inspirateur des figures 1 et 2), Patrick Bergmans, Boris Chidlovskii, Marc Dymetman, Annie Zaenen et tous nos collègues de Xerox qui ont contribué à cette réflexion sur la modélisation du document.

REFERENCES

[1] Jean-Michel Adam, *Eléments de linguistique textuelle - théorie et pratique de l'analyse textuelle*, 1990.

[2] Jean- Michel Adam, *Les textes : types et prototypes*, Nathan Université, 1992.

[3] Adler, Gujar, Harrison, O'Hara and Sellen, A Diary Study of Work-Related Reading: Design Implications for Digital Reading Devices. *CHI '98, Human Factors in Computing Systems*, Los Angeles, California, U.S.A. , Avril 1998.

[4] E. Akpotsui, V. Quint, Type Transformation in Structured Editing Systems. *Proceedings of Electronic Publishing 92 (EP92)* , C. Vanoirbeek and G. Coray, ed., University Press , Cambridge , Avril 1992.

[5] E. Akpotsui, V. Quint, C. Roisin, Type Modelling for Document Transformation in Structured Editing Systems. *Workshop on Principles of Document Processing* , Washington , Octobre 1992.

[6] J. André, R. Furuta, V. Quint, *Structured Documents* , Cambridge University Press, 1989.

[7] F. Bapst, R. Brugger, R. Ingold, Document modeling using generalized n-grams. *Proceedings of ICDAR97* , Août 1997.

[8] T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web. *Scientific American*, Mai 2001.

[9] T. Bickmore, A. Girgensohn and J. W. Sullivan, Web Page Filtering and Re-authoring for Mobile Users. *The Computer Journal* , Vol. 42 , No. 6 , 1999.

[10] T. Bickmore, B. Schilit, Digestor: Device-Independent Access To The WWW, 1997,http://www.fxpal.xerox.com/PapersAndAbstracts/papers/bic97/

[10b] Stéphane Bonhomme, Transformation de documents structurés : une combinaison des approches automatique et explicite, thèse de doctorat, Université Joseph Fourier, Décembre 1998.

[11] Allen Brown, Matthew Fuchs, Jonathan Robie, Philip Wadler, MSL: A model for W3C XML Schema. *WWW10*, Hong Kong, Mai 2001.

[12] M. Buckland, What is a digital document ? *Document numérique*, 02/1998.

[13] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, M. Tommasi, Tree Automata Techniques and Applications. Octobre 1999, http://www.grappa.univ_lille3.fr/tata

[14] Gérard Dupoirier, Considérations pour une revue orientée document numérique. *Document numérique* , Vol. 1, Janvier 1997 , pages. 11-20

[15] Mary Fernandez, Jerome Simeon, Philip Wadler, An Algebra for XML Query. *FST TCS*, Delhi, Décembre 2000

[16] SGML/XML and Forest/Hedge Automata Theory,

http://www.oasis-open.org/cover/topics.html#forestAutomata

[17] R.H.R Harper, The ethnographic turn - why it has come about and how to do it , TR EPC-1996-109, Xerox Research Centre Europe, Cambridge, UK , 1996

[18] Makoto Murata, Hedge Grammars,

http://www.horobi.com/Projects/RELAX/Archive/hedge_nice.html

[19] Paolo D'Iorio, *HyperNietzche*, PUF, 2000

[20] Donald E. Knuth, *The TeXbook*, Reading, Addison-Wesley, 1984

[21] Donald E. Knuth, *The METAFONTbook*, Reading, Addison-Wesley, 1986

[22] Dongwon Lee, Murali Mani and Makoto Murata, Reasoning about XML Schema Languages using Formal Language Theory *IBM Research Center* , Almaden , Novembre 2000 , rapport technique RJ#10197, Log#95071

[23] David Levy, Document reuse and document systems *Electronic publishing*, Vol. 6(4), Décembre, 1993 , pages 339-348

[24] C. Luc, *Contraintes sur l'architecture textuelle.* Document numérique, 02/98.

[25] Sheila A. McAlraith, Tran Cao Son, Honglei Zeng, Semantic Web Services. *Cooking up the semantic Web* , *IEEE Intelligent System* , Vol. 32, No. 3, Mars-Avril 2001, pages 46-53, http://computer.org/intelligent

[26] Robin Milner, *Communication and Concurrency* , C.A.R. Hoare Series, Prentice Hall, 1989

[27] Graham Moore, Topic Maps and RDF. XML Europe 2001, http://www.xmlhack.com/read.php?item=1232

[28] Makoto Murata, Transformation of documents and schemas by patterns and contextual conditions. *Lecture Notes in Computer Science*, Springer Verlag, Vol. 1293, 1997 , pages 153-169

[29] Makoto Murata, Data model for document transformation and assembly. *Principles of Digital Document Processing (PODDP'98)* , *Lecture Notes in Computer Science, Springer Verlag* , Vol. 1481, 1998 , pages 140-152

[30] K. O'Hara, *Towards a typology of reading goals*. , rapport technique EPC-1996-107, Xerox Research Centre Europe, Cambridge Laboratory, UK , 1996

[31] O'Hara, Sellen, *A Comparison of Reading Paper and On-Line Documents*, rapport technique EPC-1997-101Xerox Research Centre Europe , UK.

[32] O'Hara, Smith, Newman, Sellen, Student Readers' Use of Library Documents: Implications for Library Technologies. EPC-1998-101 Xerox Research Centre Europe , UK.

[33] Dave Pawson , *An Introduction to XSL Formatting Objects*, http://www.dpawson.co.uk/xsl/sect3/bk/index.html .

[34] Roger Pédauque, *Document : forme, signe et médium, les re-formulations du numérique*, Juillet 2003,

http://archivesic.ccsd.cnrs.fr/documents/archives0/00/00/05/11/index_fr.hml

[35] V. Quint, I. Vatton, Grif: an interactive system for structured document manipulation. *Text processing and document manipulation*, Cambridge University Press , 1986.

[36] A. Sellen and R. Harper, Paper as an Analytic Resource for the Design of New Technologies, EPC-1997-102, Xerox Research Centre Europe , UK.

[37] Richard Spinks, Brad Topol, Chris Seekamp, Steve Ims, Document clipping with annotation : How to keep the good stuff and throw out the rest., Avril 2001, http://www-106.ibm.com/developerworks/library/ibm-clip/

[38] Y. Stern, Les quatre dimensions du document. *Document numérique*, 01/1997.

[39] M. Murata, D. Lee, M. Mani, Taxonomy of XML schema languages using formal language theory, *Extreme Markup Languages* , Montreal , 2001

[40] A. Tozawa, Toward Static Type Checking for XSLT. *ACM Symposiun on Document Engineering*, Atlanta, USA , 9-10 Novembre2001

[41] G. Vignaux, A. Attali, M. Augier, P. Jardin, D. Piotrowski, M. Silberstein, Le projet colisciences, http://www.colisciences.net/

[42] J-Y. Vion-Dury, V. Lux, E. Pietriga, Experimenting with the Circus language for XML modelling and transformation. *Document Engineering,* Washington, U.S.A,2002.

[43] Philip Wadler, Two semantics of XPath, *Working Note,*

[44] Philip Wadler, A formal semantics of patterns in XSLT. *Markup Technologies* , Philadelphia , Décembre 1999.

[45] P. Wolper, The Meaning of formal. *ACM Computing Surveys*, Vol. 28, 1996.

Références techniques, produits

[46] Adobe, *Portable Document Format (PDF)* , http://partners.adobe.com/asn/developer/acrosdk/docs.html#filefmtspecs .

[47] Adobe, *Postscript Reference Manual* http://partners.adobe.com:80/asn/developer/pdfs/tn/psrefman.pdf .

[48] *Annotea Project* , W3C, http://www.w3.org/2001/Annotea .

[49] Air Transport Association, http://xml.coverpages.org/gov-apps.html#ata

[50] AvantGo Mobile Application http://avantgo.com/frontdoor/index.html .

[51] AT & T, The Technology for Scanned Documents on the Web http://dejavu.reseach.att.com .

[52] Dublin Core Metadata Initiative, Juin 2001 http://dublincore.org/ .

[53] Dublin Core Metadata Element Set - Version 1.0: Reference Description , Septembre1998 http://dublincore.org/documents/1998/09/dces/ .

[54] HyperText Markup Language, W3C http://www.w3.org/MarkUp

[55] A Reader's Guide to the HyTime Standard , ISO/IEC 10744:1992 2ème édition: http://www.hytime.org/papers/htguide.html .

[56] A technology related to topic Maps http://www.infoloom.com/tmweb.htm .

[57] Mathematical Markup Language v2.0, http://www.w3.org/TR/MathML2 .

[58] Synchronized Multimedia Integration Language,

http://www.w3.org/AudioVideo .

[59] *Quid Encyclopedia* (version 2000) http://www.quid.fr/ .

[60] Resource Description Framework , W3C http://www.w3.org/RDF .

[61] Voice Extensible Markup Language (VoiceXML) Version 2.0, W3C http://www.w3.org/TR/2001/WD-voicexml20-20011023/ .

[62] IBM WebSphere Transcoding Publisher . http://www.research.ibm.com/networked_data_systems/transcoding .

[63] XDuce Language http://www.cis.upenn.edu/~hahosoya/xduce.html .

[64] XML Forms, W3C http://www.w3.org/MarkUp/Forms .

[65] XML Linking Language (XLink). W3C, http://www.w3.org/XML/Linking .

[66] XML Encryption Syntax and Processing (Candidate Recommendation) ,W3C, http://www.w3.org/TR/2002/CR-xmlenc-core-20020304/ .

[67] Namespaces. W3C http://www.w3.org/TR/1999/REC-xml-names-19990114

[69] XML Schema Part 0: Primer . W3C http://www.w3.org/TR/xmlschema-0 .

[70] XML-Signature, W3C http://www.w3.org/TR/xmldsig-core

[71] Extensible Markup Language (v1.0). W3C , 1998, http://www.w3c.org/XML

[72] C.M. Sperberg-McQueen, Jean Paoli and Tim Bray, Annotated XML specification. *XML.com* , 1998 , http://www.xml.com/pub/a/axml/axmlintro.html

[73] XML Path Language V1.0. http://www.w3.org/TR/xpath .

[74] XML Stylesheet Language. http://www.w3.org/TR/xsl.html .

[75] XSL Formatting Objects,http://www.w3.org/TR/xsl/slice6.html#fo-section .

[76] XML Stylesheet Language Transformations, http://www.w3.org/TR/xslt.html

[77] XQuery. World Wide Web Consortium http://www.w3.org/XML/Query