HAL
open science

Documents structurés multimédia

Cécile Roisin

To cite this version:

Cécile Roisin. Documents structurés multimédia. Autre [cs.OH]. Institut National Polytechnique de Grenoble - INPG, 1999. tel-00004847

HAL Id: tel-00004847

https://theses.hal.science/tel-00004847

Submitted on 18 Feb 2004

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

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

Habilitation à diriger les recherches

spécialité

INFORMATIQUE

Présentée le 22 septembre 1999 à
L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

par

Cécile Roisin

Documents structurés multimédia

Jury

Président : Jean−Pierre Verjus

Rapporteurs : Serge Abiteboul
Dick Bulterman
Andrzej Duda

Examinateurs : Michel Adiba
Jacques Courtin
Marc Nanard
Vincent Quint

Remerciements

Je remercie tout d'abord Vincent Quint pour m'avoir accueillie dans le projet Opéra en 1991 et pour la confiance qu'il m'a accordée en 1996 en me donnant la responsabilité du projet Opéra. Les activités décrites dans ce mémoire sont pour beaucoup issues de directions de travail qu'il a initiées.

Je tiens à remercier Jean−Pierre Verjus de m'avoir fait l'honneur de présider le jury de mon habilitation. Je le remercie également de m'avoir permis de travailler à l'unité de recherche Inria Rhône−Alpes en position de délégation de l'université Pierre Mendès−France.

Je remercie Serge Abiteboul, Dick Bulterman et Andrzej Duda d'avoir accepté de rapporter sur mon travail. J'ai beaucoup apprécié en particulier les commentaires et suggestions de Serge Abiteboul qui m'ont permis d'améliorer la version finale de ce mémoire, ainsi que les questions soulevées par Dick Bulterman qui m'ouvrent de nouvelles perspectives de recherche.

Je remercie également Michel Adiba, Jacques Courtin et Marc Nanard pour avoir accepté de faire partie de ce jury.

La préparation de mon habilitation à diriger des recherches aboutit aujourd'hui grâce aux encouragements et au soutien de nombreuses personnes. Je tiens à citer en particulier Roland Balter et Muriel Jourdan. Je les remercie de m'avoir aidée à croire en ce projet.

Ce mémoire synthétise des activités effectuées le plus souvent en équipe. Je remercie donc tous mes co−équipiers qui furent nombreux en 15 ans de travail dans le projet Guide puis dans le projet Opéra. Je remercie Samer et Xavier de l'époque « avant Guide » ; Hervé, Hiep, Marie, Michel, Miguel, Sacha et Serge de l'équipe « langage Guide » ; merci enfin à toutes les personnes qui participent (ou ont participé) au « projet Opéra » : Vincent, Irène, Extase, Dominique, Laurent B., Nabil, Philippe, Stéphane, Manuel, Loay, Muriel, Maria, Laurent C., Laurent T., Frédéric B., Frédéric S., Lionel, Yves, Tien, Alain.

La qualité finale de ce mémoire doit beaucoup aux relectures attentives et critiques de Laurent Carcone, Muriel Jourdan, Joël Pleyber, Vincent Quint et Irène Vatton. Qu'ils en soient ici tous remerciés.

Je remercie également tous mes collègues du département informatique de l'IUT Pierre Mendès−France. Ils ont assumé les perturbations entraînées par mes nombreuses absences (maternité, détachement, délégation). Je remercie tout particulièrement mes collègues des équipes de technologie, de réseaux et de modélisation : Ana, Annie, Camille, Christian, Dominique, Louis, Jean−Louis, Joël et Sylvie.

Je remercie enfin toute ma « grande famille », mes parents et mes frères et sœurs ainsi que ma « petite famille » Benoît, Frédéric et Sophie.

Préambule

Ce document décrit les activités de recherche que j'ai effectuées à l'Imag et à l'Inria depuis ma thèse commencée en 1981. Après une présentation rapide de mon cursus scientifique dans la première section ci−dessous, je consacre l'essentiel de ce mémoire aux travaux que j'effectue dans le projet Opéra depuis 1991 sur les documents multimédia structurés. C'est en effet dans ce projet de recherche que j'ai pris des responsabilités d'encadrement scientifique, puis la direction du projet lui−même (en 1997). Ces travaux me permettent donc de présenter ce mémoire en vue d'obtenir une habilitation à diriger des recherches.

1 Des réseaux aux documents

Mon cursus scientifique est caractérisé par trois périodes correspondant chacune à une activité de recherche bien identifiée : *les réseaux d'ordinateurs et la spécification des protocoles de communication*, *les langages à objets pour les systèmes répartis* et enfin *les documents structurés multimédia*. Cette évolution thématique paraît suivre un fil conducteur qui part des « couches basses » pour aboutir à des applications directement accessibles aux utilisateurs humains, en l'occurrence les environnements auteur pour la création de documents. Cependant, le choix de ces activités a surtout été le fait d'opportunités scientifiques qu'il m'a été possible de saisir : la première a été la création du projet Guide en 1984 au moment de la fin de ma thèse et la deuxième une proposition de détachement à l'Inria pour travailler dans le projet Opéra (les projets Guide et Opéra faisant partie du même laboratoire Bull−Imag). Aussi, je ne vais pas ici chercher à mettre en perspective ces trois thèmes de recherche, mais plutôt décrire chacun d'eux en situant leur contexte et leur bilan, notamment pour expliquer mon insertion dans le projet suivant.

Enfin, il est à noter que la plupart de ces travaux ont été menés dans le cadre d'un statut d'enseignant−chercheur ; j'ai d'abord été assistante puis maître de conférences à l'université Pierre Mendès−France dans le département informatique de l'IUT 2. Cependant, sur les 8 ans d'activités dans le projet Opéra, j'ai pu être déchargée de mes tâches d'enseignante pendant 5 ans, ce qui m'a plus facilement permis de prendre des responsabilités dans ce projet.

1.1 Réseaux d'ordinateur et la spécification des protocoles de communication (1985−1991)

J'ai étudié d'une part les problèmes de spécification des protocoles dans le cadre de ma thèse intitulée « *La description des protocoles de communication avec le langage CSP* » (une publication [Roisin 1985]), et d'autre part j'ai participé à une étude sur la connexion des systèmes répartis locaux avec les réseaux ISO (une publication [HR 1986]).

Ces travaux ont été menés en parallèle avec mes activités pédagogiques au département informatique de l'IUT Pierre Mendès−France de Grenoble où j'ai eu la charge d'organiser l'enseignement sur les réseaux d'ordinateur.

Le bilan que je peux tirer de ces premières années est l'acquisition d'une première expérience de recherche fortement couplée à des activités d'enseignement ainsi que des bases solides dans le domaine des réseaux qui ont facilité mon intégration dans le projet Guide.

1.2 Langages à objets pour les systèmes répartis (1985−1991)

L'objectif du projet Guide, dirigé par Roland Balter et Sacha Krakowiak, était la conception et la réalisation d'un système réparti sur réseau local (appelé système Guide) pour le développement d'applications mettant en œuvre des traitements coopératifs, les applications visées étant le génie logiciel et la gestion de documents.

Les thèmes de recherche abordés dans ce projet couvraient les principes de conception et les techniques de réalisation des systèmes d'exploitation répartis, les méthodes de structuration et de programmation d'applications réparties, les langages à base d'objets pour l'expression de ces applications, ainsi que le support pour le développement et l'exécution des applications.

Pendant la première phase du projet, j'ai essentiellement participé à la définition du modèle d'objets et du langage Guide [Balter 1991]. Le langage Guide est le langage spécifique du système Guide permettant l'écriture d'applications réparties. Il est caractérisé par un typage fort des objets dans lequel l'interface d'accès aux objets est séparée de la définition des classes pour faciliter le contrôle statique et les multiples réalisations d'une même interface. D'autre part, il permet d'encapsuler la persistance et la répartition, déchargeant ainsi le concepteur d'application du souci des traitements spécifiques liés à la répartition. Le langage comprend également des opérateurs de parallélisme synchrone et de contrôle de l'accès aux objets partagés [DDRRR 1991].

Ces principes de programmation se retrouvent maintenant au cœur des langages modernes comme Java. Un aspect original du langage Guide a été l'utilisation d'une relation de conformité entre les types qui offre une plus grande souplesse dans la programmation des applications. Cette problématique, qui consiste à trouver des solutions optimales entre contrôle strict et flexibilité, se retrouve également dans les travaux que j'ai menés sur les transformations de structures pour les documents (voir section 4.1.2).

Pendant la deuxième phase du projet, mes activités se sont intégrées dans le thème de l'environnement d'utilisation et de programmation du système Guide et plus particulièrement sur la spécification d'un gestionnaire de types et d'un observateur graphique d'applications [JRS 1991].

Bilan

Participer à un projet de l'ampleur du projet Guide est à la fois très formateur et très exigeant, notamment parce qu'il était particulièrement difficile de suivre les activités des différentes parties du projet. Pour ma part, j'ai pu apprendre à travailler en équipe, et, sur le plan scientifique, les connaissances acquises sur les langages à objets m'ont préparée à comprendre les besoins de modélisation et de traitement des documents électroniques (mécanisme d'héritage de propriétés entre types hiérarchiquement organisés, conformité des instances vis−à−vis de types définis). Enfin, une grande partie des travaux de Guide ayant été menée dans le cadre du projet européen Comandos, j'ai eu l'opportunité de travailler avec des équipes scientifiques portugaises, italiennes et irlandaises.

1.3 Documents structurés multimédia (depuis 1991)

Par rapport aux thèmes de recherche précédents, le projet Opéra aborde des problèmes plus proches des applications et des utilisateurs. C'est un avantage lorsqu'il s'agit d'identifier de nouveaux besoins et de démontrer les résultats, contrairement au domaine des systèmes répartis pour lequel il est toujours difficile de réaliser des prototypes qui soient démonstratifs. Dans le projet Opéra, nous étudions les modèles de représentation des documents (voir section 4) dont l'objectif est de permettre de réaliser des applications couvrant tous les besoins de traitement dans la chaîne documentaire : conception, publication, archivage, réutilisation, interrogation, etc.. La validation de ces modèles est effectuée par la réalisation d'applications prototype, principalement orientées vers des fonctions d'édition (voir section 5). C'est un domaine à la fois plus spécialisé, dans lequel les usages jouent un rôle important, et qui se

trouve à la charnière de plusieurs domaines de l'informatique, comme le génie logiciel, les interfaces homme−machine ou le système (cf. section 3.2).

Depuis 1991 où j'ai rejoint le projet Opéra, j'ai pu me familiariser avec de nombreux problèmes liés au traitement électronique des documents structurés, et plus précisément :

- Représentation de la structure logique et de la structure spatiale, ou comment intégrer des structures spatiales et logiques de façon à formater les documents structurés selon des modèles de pages complexes.
- Représentation abstraite des types de documents de façon à permettre de transformer les structures de documents.
- Représentation de l'information temporelle des documents multimédia qui s'intègre aux représentations logiques et spatiales et qui soit adaptée à la construction d'environnements d'édition.

Ces différents thèmes sont développés dans la suite de ce mémoire.

2 Plan de ce mémoire

Les chapitres qui suivent sont consacrés à la description de mes activités dans le domaine des documents structurés multimédia. Ces activités se déroulent dans le projet Opéra. Depuis sa création en 1990, le projet Opéra de l'Inria Rhône−Alpes s'est intéressé à la modélisation des documents électroniques, a proposé des modèles et a mis en œuvre des environnements d'édition qui valident ces modèles [Opéra 1997], [Inria 1998]. Avec l'intégration de données temporisées comme des éléments vidéo, sonores ou des objets animés, les documents électroniques sont maintenant de nature multimédia et de nouveaux modèles sont définis pour prendre en compte cette dimension temporelle des documents : un modèle de spécification à base de contraintes temporelles a été proposé par le projet Opéra et il est expérimenté dans un environnement d'édition/présentation de documents multimédia [JLRST 1998].

Je ne décris pas dans ce mémoire mes travaux selon un ordre chronologique mais plutôt selon une approche induite par les modèles de documents et les applications qui ont servi de cadre à ces activités. Ainsi, j'ai organisé la suite de ce document en 3 chapitres principaux :

1. **« Qu'est−ce qu'un document ? »** (chapitre 3) qui constitue une introduction à la notion de document et aux problèmes liés au traitement électronique des documents.
2. **« Modèles de documents »** (chapitre 4) dans lequel sont décrits les modèles qui permettent de prendre en compte certaines des caractéristiques des documents. Ce chapitre définit brièvement les concepts, décrit les standards correspondants et surtout situe les modèles qui ont été proposés dans le cadre de mes travaux. Je présente en particulier mes contributions : le modèle des types de documents qui est à la base des travaux sur les transformations de structures, le modèle de spécification de pages et le formateur associé qui m'ont permis d'améliorer le formatage des documents structurés et enfin le modèle de composition temporelle à base de contraintes qui est au cœur des travaux du projet Opéra sur les documents multimédia.
3. **« Applications documentaires »** (chapitre 5) qui présente des applications de traitement de documents et plus précisément les applications auteur développées dans le projet Opéra : édition de documents structurés, transformation de structures et édition de documents multimédia. Je situe là aussi ma contribution et les travaux que j'ai encadrés dans ces différentes classes d'application.

Ce découpage présente l'avantage de séparer clairement les contributions sur les modèles de celles sur les outils. Par contre, certains résultats de modélisation trouvent leur pleine validation dans les outils qui les utilisent mais ne sont pas décrits à la suite. Pour aider le lecteur à se repérer, voici une grille récapitulative des chapitres 4 et 5.

	structures logiques	structures spatiales	structures temporelles	structures de navigation
Modèles	section 4.1	section 4.2	section 4.3	section 4.4
Outils	sections 5.1 et 5.2		section 5.3	sections 5.2 et 5.3

La plupart des activités décrites ici sont le fruit d'un travail d'équipe avec les différentes personnes qui font partie (ou ont fait partie) du projet Opéra. Les signatures multiples des publications auxquelles j'ai contribué reflètent ce mode de travail. L'usage de la première personne du pluriel employée fréquemment dans ce mémoire correspond donc effectivement à des participations de plusieurs personnes dans les choix, les spécifications et les réalisations décrites. Cependant, je m'efforce dans la mesure du possible d'identifier ma part personnelle dans ces travaux.

3 Qu'est−ce qu'un document ?

De par son origine étymologique (de « docere », instruire), le terme « document » est porteur du sens « transmission de connaissance, de savoir, d'information ». Cette capacité de transmission est obtenue grâce à la mise sous une forme persistante de cette information : des tablettes d'argile utilisées en Mésopotamie jusqu'aux cédéroms utilisés aujourd'hui en passant par tous les supports employés pour les livres (papyrus, parchemin, papier) [Blasselle 1997].

Selon cette approche, le document est une forme pérenne (en général sous forme textuelle) d'une information. Un document est conçu par un auteur dans le but d'être reçu par un ou plusieurs lecteurs.

Cependant, ce schéma « *auteur* <−− *document* −−> *lecteur* » peut être décliné selon toute une palette de possibilités en fonction des transformations qui peuvent être apportées à l'information initiale fournie par l'auteur (sans parler de l'interprétation sémantique que tout lecteur fait de tout message) : rôle des éditeurs qui choisissent la mise en forme de l'information, rôle des traducteurs qui choisissent le mode d'expression de l'information dans une autre langue que celle de l'auteur, rôle du lecteur dans son mode d'accès à l'information. Par exemple, dans le cas des documents électroniques, la qualité de présentation du document dépend du système de restitution : réseau, puissance du processeur, qualité de l'écran, des haut−parleurs, etc. De plus, la dynamique et l'interactivité des documents, du fait de leur contenu et des outils qui permettent d'y accéder, permettent aux lecteurs d'être de plus en plus actifs et rendent en conséquence de plus en plus difficile l'identification de l'information initiale fournie par l'auteur, c'est−à−dire « le document ».

Dans cette section, j'illustre tout d'abord cette difficulté en présentant le document selon plusieurs facettes. Puis je présente quelques problèmes liés au traitement électronique des documents.

3.1 Quelles définitions ?

À l'ère des hypermédia, du web, des cédéroms, la notion de document devient difficile à identifier car on ne peut plus se référer à des repères précis. En effet, différentes caractéristiques, parfois contradictoires, peuvent être considérées pour établir une définition du document. En voici ci−dessous quelques unes, chacune couvrant partiellement la notion de document.

Contenu : « un document est un regroupement d'objets »

Cette définition permet de caractériser les documents par leur contenu. Le terme objet doit ici être pris dans le sens d'une entité élémentaire manipulée par les applications, comme les éléments textuels ou graphiques, les images, voire les sons ou les vidéos. Outre la difficulté à définir le niveau adéquat de granularité de ces objets élémentaires, cette définition ne permet pas de prendre en compte le cas des documents engendrés dynamiquement par une requête sur une base de données qui évolue elle−même dynamiquement.

Espace physique : « un document est un défini par son support physique »

C'est le cas « classique » du livre présenté sur support papier sous la forme d'une suite de pages et identifié par un numéro unique fourni par l'ISBN. Il est clair que cette définition n'est pas applicable à l'information disponible sur le web : du fait des liens et des ressources attachés à un document, on ne peut plus se fier aux limites physiques fixées par le livre (ou d'un équivalent électronique qui pourrait être le fichier) pour parler de document. On se rapproche cependant de la notion traditionnelle de document dans le cas où le support électronique représente l'information sous forme d'un codage en pages numériques (Postscript, pdf) ou de bitmaps de pages scannées.

L'identification fournie par l'organisation du système de stockage dans laquelle l'information est conservée est un autre moyen de caractériser un document (avec les limites évoquées précédemment) : code de bibliothèque, nom de fichier, URI des informations du web, etc.

Organisation logique : « un document est un ensemble d'objets organisés hiérarchiquement »

Il s'agit là d'identifier des composants intermédiaires qui définissent des regroupements logiques d'objets (l'ensemble des paragraphes formant une section par exemple) comme décrit en section 4.1. Cependant, cette approche doit être élargie pour couvrir les besoins hypertexte pour lesquels une organisation de liens doit être ajoutée à l'organisation hiérarchique des informations.

Par exemple, on peut utiliser les liens comme support de relations sémantiques entre composants ou pour fragmenter les informations documentaires de grande taille (exemple : une thèse éclatée en un ensemble de pages HTML accessibles sur le web). Il est alors difficile de fixer *a priori* une frontière

dans de tels réseaux de liens. La toile du web n'est−elle donc pas alors un gigantesque document ?

Date : « un document est un ensemble d'informations datées »

Pour prendre en compte l'évolution des informations contenues dans un document, on peut les dater. Dès lors que ces informations peuvent évoluer, il faut gérer les relations entre les documents relatifs à un même ensemble d'informations : on utilise alors des protocoles comme WebDAV [W3C WebDAV 1998] et des systèmes de gestion de versions. Ces outils n'ont cependant pas de sens dans le cas où les informations ne sont pas stables (lorsqu'elles sont produites dynamiquement).

Scénario temporel : « un document est un enchaînement temporel d'objets »

Cette définition caractérise les présentations multimédia dans laquelle des objets média (son, texte, image, vidéo) sont organisés temporellement.

Traitement : « un document est un objet défini par des données et des traitements »

Un document peut être non seulement défini par un contenu mais aussi par les traitements qu'il est possible d'y appliquer (rédaction, lecture, navigation, validation, ajout d'annotations) ainsi que par les acteurs humains ou logiciels qui peuvent agir sur le document. Ce type de caractérisation peut être fonction des étapes de la vie du document, des parties de document concernées, des actions à effectuer, des rôles attribués aux acteurs (lecteur ou écrivain par exemple) ou de l'identité des acteurs (identification par nom, par appartenance à un groupe, par localisation physique, etc.). L'utilisation d'informations complémentaires au document lui−même est alors nécessaire [W3C RDF 1998].

Dans la suite de ce mémoire, les documents sont considérés comme des entités constituées d'un ensemble d'éléments d'information de base reliés par des relations de différentes natures (relations de composition, spatiales, temporelles et de navigation). Les propriétés et les relations qui sont ainsi attachées aux éléments de documents permettent de couvrir un grand nombre des besoins évoqués dans les définitions précédentes. La section 4 est consacrée à l'étude de ces relations et des caractéristiques des éléments de base.

3.2 Quels sont les problèmes liés aux traitements électroniques des documents ?

Les multiples facettes présentées dans la section précédente que l'on peut attacher aux documents constituent une difficulté pour la mise en place de modèles généraux qui intègrent les différents besoins de traitements des documents. C'est donc en premier lieu un travail de modélisation qu'il faut effectuer pour représenter les documents (voir chapitre 4). On est ainsi passé des représentations sous forme de simples chaînes textuelles à des modèles riches à base de multiples structures d'arbres, de graphes, etc. Afin de situer les travaux présentés dans les chapitres 4 et 5, je décris ci−dessous le contexte dans lequel il s'intègre que ce soit sur le plan économique, scientifique ou applicatif.

Contexte des travaux sur les documents électroniques

Les recherches menées dans le domaine des documents électroniques doivent tenir compte d'un contexte technologique et économique contraignant que l'on peut caractériser par :

- Les problèmes scientifiques intrinsèques au domaine comme la nature des informations traitées et le type des applications (applications interactives, destinées à des utilisateurs non informaticiens).
- Un contexte de standardisation fort qui impose de tenir compte des standards pour que les solutions proposées aient une certaine crédibilité.
- La situation du marché des logiciels de traitement de documents qui rend difficile l'émergence de nouvelles approches et de nouveaux usages dans ces applications.
- Enfin, mais ceci est vrai de beaucoup de domaines de l'informatique, la rapidité d'évolution des techniques, et en particulier des réseaux et du multimédia, qui remet en cause des solutions existantes que ce soit pour la représentation des données, pour les choix d'architecture logicielle ou pour les interfaces des applications.

Relations avec d'autres domaines

Les travaux sur les documents électroniques sont en relation avec de nombreux autres domaines de recherche qui vont des travaux en psychologie cognitive et ergonomie jusqu'aux systèmes et réseaux.

Le document (électronique ou non) étant à la base un support de communication humaine, les travaux sur la psychologie cognitive et l'ergonomie sont nécessaires pour en définir les bons modes de représentation et les outils de traitement. Par exemple, des études [Montarnal 1997] faites sur l'activité de rédaction ont montré que les rédacteurs étaient confrontés à la contrainte de linéarisation : la description mentale des objets est multidimentionnelle tandis que le mode de création, à travers le langage, est monotâche. Il est alors intéressant d'offrir aux rédacteurs des moyens de communiquer la complexité de la structure des objets décrits (hypertexte, figures, etc.) et d'évaluer ces moyens auprès des lecteurs [Bétrancourt 1996]. Avec l'introduction du multimédia, ces travaux s'orientent vers la perception de l'information temporelle : comment concevoir et optimiser le traitement de la dimension temporelle des documents multimédia à la fois du point de vue du rédacteur (quelles sont les représentations mentales de la dimension temporelle) et du point de vue du lecteur (quels sont les facteurs qui influencent la compréhension d'un document ayant une dimension temporelle) [BTP 1998].

Les travaux en interfaces homme−machine intéressent particulièrement notre domaine puisqu'un des résultats attendus des recherches sur les documents électroniques est de proposer des outils de traitement de documents. Non seulement les résultats en IHM sont importants pour la définition d'applications documentaires, comme par exemple les travaux sur la visualisation d'informations hiérarchiques, multidimensionnelles et/ou volumineuses (arbres coniques et hyperboliques, murs en perspectives, etc.), mais inversement, le document peut être un support pour la communication entre applications et utilisateurs [QV 1994]. Un aspect particulier à considérer lors de la conception d'interfaces pour les applications de traitement de documents est la nécessité de tenir compte des comportements déjà ancrés chez les utilisateurs. En effet, l'utilisation quotidienne des outils de traitement de documents induit des habitudes et des repères dont il faut tenir compte lors de la conception de nouvelles applications.

Le génie logiciel est un autre domaine ayant d'étroites relations avec les documents : il est classique de considérer un document comme un programme (qui respecte une grammaire) [AF 1997]. De même, les traitements qui s'appuient sur le typage fort des documents ont fait émerger des besoins de transformations de documents, par exemple lors d'opérations de copier−coller, qui sont proches de besoins similaires dans les logiciels : coercition de types, gestion de configurations, évolution de schémas. Le processus de développement d'une grosse documentation est également à rapprocher de celui d'un gros logiciel de par les besoins en gestion de versions et de configurationou en maintien de cohérence de données.

Les travaux sur les bases de données touchent de près ceux des documents : les besoins de représentation des données sont proches dans ces deux domaines. Ainsi on constate que, de plus en plus, le standard XML [W3C XML 1998], qui est issu du domaine des documents, se trouve au cœur des environnements de bases de données. Les bases de données peuvent être utilisées comme contenants de données documentaires (comme la gestion de fonds documentaires hypertexte pour la documentation technique aéronautique [François 1997]), ce qui amène à l'étude de nouveaux champs d'investigation comme :

- Les bases de données semi−structurées (comme le sont les documents) : on peut citer le système Lore avec son langage d'interrogation Lorel [MAG-QW 1997], ainsi que le langage d'interrogation de documents structurés SgmlQL [LMR 1998].
- Les bases de données multimédia [AN 1997]. Par exemple, STORM est un modèle pour la gestion de données temporelles [AM 1997].
- Les langages d'interrogation pour le web, et en particulier pour les données spécifiées en XML : de nombreux travaux sont en cours sur le sujet, comme la proposition XML−QL soumise au W3C [W3C XML−QL 1998], le langage XQL (*XML Query Language*) ou le langage Lorel pré−cité (voir les propositions discutées lors du workshop QL'98 [Marchiori 1998]).

Enfin, le système et les réseaux permettent d'aborder des aspects fondamentaux nécessaires à la construction des applications de traitement de documents. Dans le cas d'applications interactives, il est nécessaire de choisir des solutions efficaces de gestion d'événements et de ressources. Le besoin de traiter efficacement des données consommatrices de puissance de calcul (comme pour la décompression de l'audio et la vidéo) amènent à des solutions construites selon des architectures multiprocessus dans lesquelles des techniques d'ordonnancement et de synchronisation sont utilisées. Enfin les applications sont construites sur des infrastructures en réseau, que ce soit pour accéder à des données réparties ou pour coordonner des traitements répartis comme dans le cas de l'édition coopérative. Il est de plus en plus vrai que l'édition et la consultation de documents s'effectue dans le cadre du web.

Les axes de recherche actuels

Deux mots−clés résument les deux axes (non−disjoints) dans lesquels l'activité sur les documents électroniques a pris récemment un essor très important : le web et le multimédia.

L'essor formidable du web a entraîné (et parfois écrasé!) dans son sillage de nombreux domaines liés aux documents comme l'hypertexte, les documents structurés ou l'indexation et la recherche d'information. Les groupes de travail du consortium W3C constituent des lieux privilégiés où sont discutés, spécifiés et expérimentés les standards qui sont et seront les références du domaine. Ainsi en est−il du standard XML qui peut être considéré comme le standard pivot du web à partir duquel et autour duquel les autres standards de représentation des informations du web sont définis. Ces travaux constituent l'infrastructure de base pour la définition de nouvelles applications réparties qui s'appuient sur le concept de document pour présenter les données et interagir avec le (ou les) utilisateur(s).

Le multimédia prend une importance de plus en plus grande dans notre société, notamment par le fait que la communication se fait de plus en plus en utilisant des supports qui intègrent du son, de la vidéo, du texte, etc. Les cédéroms, les sites web, les bornes interactives ou la télévision interactive ne sont plus maintenant considérés comme des technologies futuristes mais entrent dans le quotidien des entreprises et du grand public. L'émergence à grande échelle de ces applications multimédia implique d'apporter des solutions pour la création et pour la présentation de ces applications sur des terminaux aussi divers que des PC, des screenphones, des PDA (Portable Digital Assistant) ou même des téléphones cellulaires. La définition de formats standards est bien sûr au cœur de ce domaine puisqu'ils doivent répondre aux besoins de portabilité, d'échange, de pouvoir d'expression et de faisabilité. Les travaux actuels menés dans le cadre du consortium W3C sont essentiels pour l'essor de ce domaine.

Domaines d'applications

Jusqu'à très récemment, les résultats des travaux de recherche sur les documents ont principalement suscité l'intérêt des secteurs comme la documentation technique ou les métiers du livre dans lesquels les volumes d'information sont importants et pour lesquels l'échange de documents sous une forme électronique exploitable est nécessaire. Ainsi des solutions à base de formats standard comme SGML ou HyTime ont−elles émergé. Elles reposent sur une structuration et un typage fort des données documentaires. Un exemple caractéristique est l'aéronautique qui utilise des

techniques fondées sur SGML dans ses chaînes documentaires depuis la création et la gestion de fonds documentaires jusqu'à la diffusion des documents techniques à leurs clients grâce à la définition de DTD communes comme celles de l'ATA (*Air Transport Association*). Cependant, si des solutions à base d'hypermédia commencent à être employées pour permettre la mise en ligne de ces documents, ces secteurs ne sont pas encore prêts à introduire massivement le multimédia pour des raisons liées au mode de fonctionnement de ce marché qui s'appuie sur des standards, au volume d'informations à gérer et à un important existant à prendre en compte [DRTV 1999].

Parmi les secteurs qui utilisent la technologie du multimédia et plus précisément la notion de document multimédia, on citera deux d'entre eux, qui sont parmi les plus concernés actuellement : l'enseignement assisté par ordinateur (*EAO*) et la médecine. L'EAO tire parti des caractéristiques de différents média pour réaliser des supports pédagogiques qui soient d'une part attractifs grâce aux images, aux animations et aux sons, et d'autre part plus interactifs et adaptables aux élèves grâce aux fonctions de navigation hypermédia [Stemler 1997]. De même, les données issues des plateaux techniques d'imagerie médicale, comme les images par rayons X, par résonance magnétique ou par échographie, peuvent être exploitées sous forme numérique et intégrées à des données textuelles (par exemple les informations relatives au patient), et sonores (les commentaires du médecin), pour former de véritables documents multimédia médicaux qui peuvent être consultés à distance par les médecins.

4 Modèles de documents

Depuis de nombreuses années, les documents électroniques ont fait l'objet d'études qui ont conduit à l'identification de caractéristiques attachées aux documents et classées selon différentes dimensions [AFQ 1989], [Stern 1997], [ISO 1996]. Le résultat majeur de ces travaux est la définition de standards comme XML (*Extended Markup Language*) [W3C XML 1998] qui permettent de représenter la *dimension logique* des documents indépendamment de leur contenu et de leur *dimension spatiale* (forme), cette dernière faisant elle−même l'objet de standards comme DSSSL, CSS ou XSL. Cette décomposition de l'information portée par les documents a pour premier objectif de faciliter la portabilité des documents ainsi que leur traitement par des applications variées. Sur la base de cette approche, d'autres caractéristiques sont regroupées pour former de nouvelles dimensions : la *dimension hypertexte*, correspondant à l'ensemble des informations permettant de lier des documents (ou des fragments de documents) entre eux, et la *dimension temporelle* qui identifie le comportement dans le temps des documents (comme la durée des objets ou leur synchronisation).

Pour chacune de ces dimensions, le travail de modélisation consiste à identifier d'une part les entités de base, comme les éléments textuels pour la structure logique (#PCDATA dans la syntaxe XML), les boîtes pour le placement spatial, les intervalles temporels pour le temps, et d'autre part leur mode de composition qui doit permettre de couvrir des besoins d'expression comme :

- La composition logique : « un livre est constitué d'un titre, d'un auteur et d'une suite de chapitres, chacun d'eux étant formé d'un titre et d'une suite de paragraphes ».
- La composition spatiale, appelée également composition graphique : « le titre occupe une largeur de 80% de celle de la page et est centré ; chaque note doit être placée en bas de la page dans laquelle se trouve la première référence à cette note ».
- La composition temporelle : « la présentation de l'entreprise débute par l'affichage de son logo pendant 5 secondes, puis la photo du directeur est affichée en même temps que son discours est joué ; la fin de la présentation est composée d'une vidéo de 3 minutes ».
- La composition hypertexte : « un lien est placé entre chaque entrée bibliographique et chacune des références à cette entrée ».

La diversité des besoins en matière de représentation de documents a conduit à une approche modulaire de ces spécifications et à la notion de *profils* (sous−ensemble de modules de spécifications pour un domaine applicatif donné). C'est une approche en cours de discussion dans le groupe de travail « *XML Syntax* » du W3C.

Notons également qu'un des objectifs des travaux de modélisation des documents est de faciliter la réutilisation, d'où la notion de modèles génériques que l'on retrouve pour les dimensions logique et spatiale (et dans une plus faible mesure pour l'organisation hypertexte).

Ce chapitre est structuré à partir de cette analyse des documents et comporte donc quatre parties, chacune correspondant à une des dimensions introduites ci−dessus. Pour chacune d'elles, j'en rappelle les principes et les travaux principaux, puis je décris la contribution que nous avons apportée.

4.1 Modèles de représentation des structures logiques

4.1.1 Principes et langages

La représentation de la structure logique des documents s'appuie sur 3 entités :

- Les éléments de base non décomposables qui constituent le contenu.
- Les éléments composites obtenus par composition d'éléments de base ou d'éléments composites ; les opérateurs de composition sont l'identité, l'agrégation, la liste et le choix .
- Les attributs qui peuvent être associés aux éléments pour leur adjoindre des informations sémantiques.

Ces principes permettent d'organiser les documents selon une structure hiérarchique (voir figure 1) d'éléments typés, décrite linéairement comme un emboîtement de marques (ou balises) encapsulant les éléments de base. Ainsi le document de la figure 1 est représenté par : <livre> <titre> Madame Bovary </titre> <auteur> Flaubert </auteur> <listeChapitres> <chapitre> <titreChap> xxxx </titreChap> ...

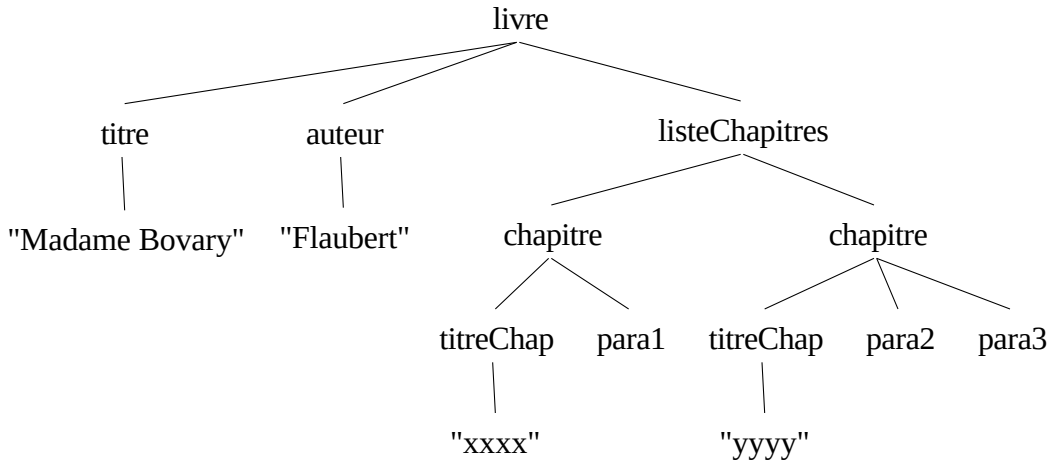


Figure 1 : Structure logique d'un livre

Des modèles génériques de description de structures logiques de documents permettent de décrire des classes pour les documents qui partagent les mêmes spécifications. Ces classes sont définies sous forme de grammaires qui peuvent être représentées par des graphes de type. Par exemple, avec la syntaxe XML, un modèle de document (ou DTD pour *Document Type Definition*) pour une classe de livres, pourrait être :

```
<!ELEMENT livre (titre, auteur, listeChapitres) >
<!ELEMENT listeChapitres     (chapitre)+        >
<!ELEMENT chapitre  (titreChap, para+)          >
<!ELEMENT para  (listeItem | simplePara)        >
<!ELEMENT listeItem  (para)+                    >
<!ELEMENT (titre | auteur | titreChap |
           SimplePara)     (#PCDATA)            >
```

Figure 2 : une DTD pour les livres

De nombreux secteurs professionnels définissent des DTD qui sont à la base des échanges de documents entre les acteurs de ces domaines : CALS pour la documentation technique des armements, ATA pour la documentation technique aéronautique, TEI (Text Encoding Initiative [TEI 1998]) pour les documents littéraires, ISO 12083 et MathML [W3C MathML 1998] pour les documents scientifiques. Des travaux sont en cours pour la définition de DTD pour les graphiques 2D [W3C SVG 1998].

Les avantages de l'utilisation de tels modèles ont été identifiés depuis longtemps [FQA 1988], mais, malgré la définition il y a plus de 10 ans de standards comme SGML [ISO 1986] ou ODA qui appliquent ces principes, les formats de

représentation de documents utilisés par la majorité des outils actuels sont des formats propriétaires avec peu de structuration de l'information. Cette situation peut s'expliquer par l'état du marché et par le peu d'outils simples et peu coûteux permettant de traiter des documents structurés. La rigidité et la complexité induites par certains choix de base de SGML ont été également des facteurs de frein au déploiement d'outils adaptés à cette norme.

Le besoin d'échanger à travers le web des informations sémantiquement plus riches que ne permet de le faire HTML a conduit les acteurs concernés au sein du W3C à la définition d'un nouveau standard issu de SGML : XML [W3C XML 1998]. Pour répondre aux besoins spécifiques du web, XML ainsi que les standards associés offrent :

- des simplifications syntaxiques (par rapport à SGML),
- la définition de structures de lien et d'adressage plus riches (XLink [W3C XLink 1999] et XPointer [W3C XPointer 1998]),
- des possibilités de modularité grâce à la spécification d'espaces de noms [W3C Namespaces 1999],
- des possibilités d'association de contraintes syntaxiques ou sémantiques (XML Schéma [W3C XML Schema 1999]),
- la possibilité de traiter un document XML même si l'on ne dispose pas de la DTD correspondante (notion de document bien−formé).

Il est clair que nous sommes à une étape charnière dans ce domaine car la dynamique générée autour de XML concerne non seulement les navigateurs web (Netscape, Microsoft) ou les outils SGML (Softquad, Arbortext, Omnimark), mais aussi la bureautique (Microsoft, Lotus), les outils graphiques, les SGBD (Oracle, SAP) et les outils de développement (SUN, IBM).

4.1.2 Modèle de type de documents pour les transformations de structures

Le projet Opéra s'est intéressé très tôt aux modèles pour la représentation des documents structurés : les premiers travaux remontent à 1985 et, dans sa thèse [Quint 1987], V. Quint décrit les langages S, P, et T qui sont au cœur des systèmes prototype développés par le projet. Le langage S (S pour structure) offre un pouvoir de description des structures génériques de documents très proche de celui de SGML avec cependant quelques principes qui se sont avérés très pertinents : ainsi en est−il de la possibilité de définir des modèles de façon modulaire et de la notion de schéma d'extension.

Pour ma part, j'ai participé aux travaux sur l'analyse des besoins en matière d'évolution des DTD et de transformation de documents structurés. Cette activité a démarré dans le cadre de la thèse d'E. Akpotsui (1990−1993) [Akpotsui 1993] sur les *transformations statiques* de bases documentaires qui permettent de faire évoluer des documents d'une base lorsque les DTD évoluent. Les principaux résultats de cette thèse sont d'une part une analyse des différents types de transformations élémentaires qui peuvent survenir lors du changement de structure d'un document [AQ 1992], et d'autre part la spécification et la mise en œuvre d'un outil de transformation statique de documents qui s'appuie sur cette analyse pour comparer les DTD. J'ai participé à l'encadrement de la fin de cette thèse et j'ai plus particulièrement travaillé avec E. Akpotsui sur les *transformations dynamiques*, c'est−à−dire celles qui surviennent au cours d'une session d'édition. Nous avons montré que de telles transformations ne pouvaient s'appuyer sur les mêmes principes (langage d'expression des couplages et/ou transformations fondées sur l'homonymie des types) mais qu'une approche plus automatique était nécessaire [RCA 1997]. Cet axe de recherche s'est poursuivi dans le cadre du stage d'ingénieur Cnam de P. Claves [Claves 1995], puis par la thèse de S. Bonhomme [Bonhomme 1998] que j'ai encadrés.

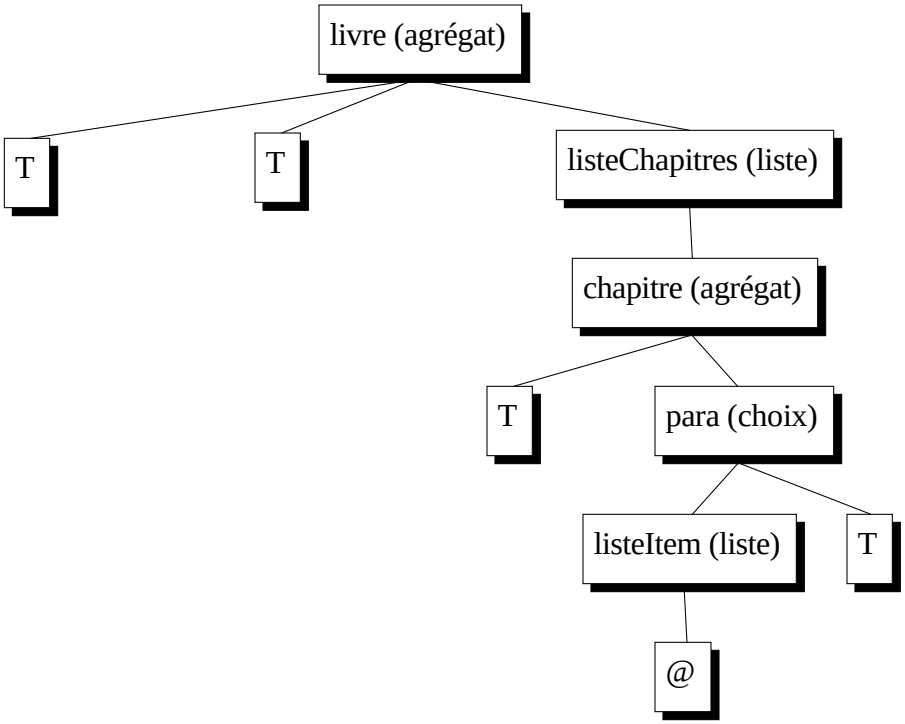
Plus précisément, nous avons proposé une représentation des structures logiques génériques (ou *types*) de documents qui permette de les comparer. Cette comparaison est à la base du processus de transformation automatique de documents structurés décrit en section 5.1. L'objectif d'une telle représentation est d'une part de coder l'information des types jugée utile pour les besoins de la transformation, et d'autre part d'en trouver une forme qui soit efficace pour la mise en œuvre d'algorithmes de comparaison. Dans la modélisation choisie, trois informations sont retenues pour la comparaison : l'organisation hiérarchique des types, les constructeurs de types et les types de base [AQR 1997]. Le processus de modélisation comprend quatre étapes (voir la thèse de S. Bonhomme pour une description détaillée [Bonhomme 1998]) :

- Mise sous forme canonique des graphes de types pour en produire des *arbres de types* (par renommage des types et élimination des identités et de la récursion). Les types de base (texte, image) sont des feuilles étiquetées par un symbole T pour texte, I pour image, etc. Les sous−arbres correspondant à des récursions sont représentés par un nœud terminal étiqueté avec le symbole @.
- Décoration des arbres de types avec les constructeurs de chaque composite.

- Ordonnancement des types d'un même ascendant. L'ordre choisi doit permettre d'obtenir des résultats de comparaison qui soient les plus pertinents pour l'application de transformation.
- Linéarisation des arbres pour obtenir des *empreintes de types*, cette dernière opération étant motivée par des raisons de performances pour appliquer des comparaisons sur les chaînes et non sur des arbres. Une empreinte est une expression parenthésée telle que chaque couple de parenthèses utilisé pour un nœud corresponde au constructeur du type de ce nœud :

Constructeur	*Liste*	*Choix*	*Agrégat*
Caractères	**( )**	**[ ]**	**{ }**

À titre d'illustration, l'arbre de types et l'empreinte correspondante pour la DTD *livre* donnée en figure 2 sont les suivants (figure 3) :



empreinte de livre : { T T ({ T [(@) T] }) }

Figure 3 : Arbre de types et empreinte pour la DTD livre

Les empreintes ainsi obtenues sont des mots de Dyck dont le langage est un langage algébrique, donc reconnu par un automate à pile, ce qui permet de ramener le problème de comparaison de types de documents à un problème d'intersection de langages algébriques.

Cette modélisation des types de document permet de bien représenter l'information structurale que l'on souhaite préserver le plus possible lors des transformations de documents structurés. Cependant, du fait de la généralisation des types résultant de cette modélisation (les noms de types ne sont pas conservés), le processus de transformation par comparaison d'empreintes produit le plus souvent de nombreuses possibilités de transformations et l'on tombe alors sur un problème d'orientation du processus vers la solution souhaitée. La solution que nous avons proposée est de piloter ce processus automatique avec des informations de *couplage* entre les types source et destination [BR 1999] (voir section 5.1.2).

Ces travaux sont à comparer avec ce qui est fait dans le domaine des bases de données pour permettre l'intégration de données issues de sources différentes (structurées ou non). Le projet TSIMMIS [CGHIP 1994] par exemple offre un cadre général et des outils (générateur de traducteurs et de médiateurs) pour permettre à des utilisateurs de mettre en œuvre leurs applications. Le cœur du système est un modèle d'information (Object Exchange Information) au travers duquel les différentes sources sont accédées. Le système YAT [CCS 1998] vise également à répondre aux besoins d'intégration de données en s'appuyant sur un modèle de données dont les caractéristiques permettent d'utiliser des techniques d'optimisation de requêtes, y compris lors de l'accès aux données semistructurées. Ainsi, l'objectif de ces modèles est l'efficacité dans le traitement de l'accès à l'information et de la génération des structures cibles dont la spécification est complètement décrite dans un langage *ad hoc*. Ces outils sont donc adaptés aux applications traitant de gros volumes d'information dont les structures sont connues. Par contre, le modèle que nous proposons vise à réaliser des transformations pour des structures non connues *a priori* et de faible volume (un fragment de document lors d'une opération d'édition par exemple).

4.2 Modèles de représentation des structures spatiales

La présentation spatiale (ou graphique, *layout* en anglais) des documents s'appuie d'une part sur une expression des *propriétés de style*, et d'autre part sur un *processus de formatage* dont le résultat est un document formaté (tous les éléments le constituant sont disposés physiquement sur le support de sortie). Les techniques utilisées pour réaliser le formatage sont en partie dépendantes du mode de spécification des propriétés de style : avec le mode procédural, le processus de formatage se ramène à une simple exécution des commandes spécifiées, tandis qu'avec le mode déclaratif,

les algorithmes utilisés sont plus complexes puisqu'il faut interpréter les déclarations pour produire le résultat final. Dans ce qui suit, nous ne considérons que ce dernier mode d'expression car c'est celui qui offre le plus de possibilités (cf. 4.2.1 ci−dessous). Dans le projet Opéra, nous avons expérimenté deux types d'algorithmes de formatage :

- Le formatage de l'éditeur de documents structurés Thot que j'ai étendu par une mise en page plus riche. Ce formatage s'appuie sur la spécification d'un modèle de pages et sur la structure logique du document pour calculer les contraintes de placement. Je décris ce travail en section 4.2.2.
- Le placement spatial utilisant un résolveur de contraintes dans le système d'édition/présentation de documents multimédia Madeus. Cette expérimentation est décrite en section 4.2.3.

4.2.1 Principes et langages

La spécification des propriétés de style peut être de type procédural ou déclaratif, de nature spécifique ou générique ou encore définie dans un langage propriétaire ou bien standard. L'évolution du mode de spécification de ces propriétés est caractérisée par le passage d'un mode procédural, spécifique et propriétaire, à un mode déclaratif, générique et standard, ce qui correspond à l'évolution des modèles de structures logiques de documents. En effet, dès lors que l'on manipule des documents codés selon un langage de marquage permettant de représenter leur contenu et leur organisation logique (indépendamment de l'organisation spatiale), la spécification et le codage des informations relatives à la présentation graphique de ces documents peut être effectuée de façon externe. Ainsi, à la suite de SGML, un groupe de l'ISO a défini le standard DSSSL (*Document Style Semantics and Specification Language*) [ISO 1996]. De même, dès 1995, le W3C a mis en place un groupe de travail pour la définition d'un langage de styles pour les documents HTML et en est maintenant à la deuxième version de la recommandation CSS (*Cascading Style Sheets*) [W3C CSS 1998] (la troisième est en cours). Plus récemment, le W3C, cherchant à répondre aux nouveaux besoins de présentation de documents fortement structurés (XML), a mis en place un groupe de travail pour la définition du langage XSL (*Extensible Stylesheet Language*) [W3C XSL 1999].

Comme pour les modèles de structuration logique, les modèles de structuration spatiale sont passés de modèles de flots linéaires (suite de blocs de texte appelée *galée*) à des modèles de flots arborescents (boîtes imbriquées). Ainsi, la structure spatiale

d'un document est un arbre de boîtes tel que la relation de composition définie par cette structure hiérarchique corresponde à une relation d'inclusion de surfaces. Par exemple, la figure 4 ci−dessous décrit la structure d'un document en pages, colonnes et blocs.

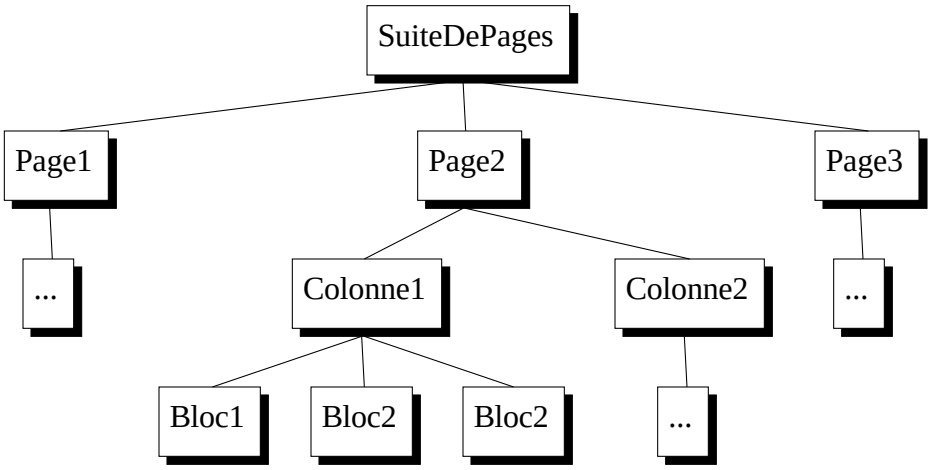


Figure 4 : Structure spatiale d'un document

Caractérisation des modèles de structure spatiale

Un modèle de spécification des structures spatiales de documents structurés est défini par :

- La spécification du jeu de propriétés de formatage (ou *style sheet*), qui caractérise le pouvoir d'expression du modèle. S'il est aisé de définir un ensemble minimal de propriétés et d'y attacher le traitement correspondant (taille des caractères, fonte, couleur, indentation, etc.), certaines propriétés peuvent être très complexes, comme le placement spatial (règles relatives aux veuves et aux orphelins, placement des notes, des figures, etc.).
- Le mode d'affectation de ces propriétés aux constituants du document à présenter.

On peut identifier trois niveaux d'affectation des propriétés de formatage :

1. Lorsque les propriétés sont attachées directement aux éléments du document, on parle de « présentation spécifique ». Les propriétés sont ainsi insérées dans le flot du contenu du document, comme les commandes TeX par exemple.
2. Si les propriétés sont attachées aux types des éléments de la structure logique, on peut définir des modèles génériques de présentation [FQA 1988]. Les langages de style (propriétaires ou standard) appartiennent à cette catégorie,

comme par exemple, le langage CSS ou le langage P du système Thot [Quint 1994]. Ces modèles s'appuient fortement sur le modèle de structure logique pour attacher les propriétés de présentation. Leur principal inconvénient est de rendre plus difficile l'expression de propriétés indépendantes des types d'éléments : c'est le cas des modèles de pages dont les caractéristiques dépendent du support de sortie. De plus, pour assurer l'héritage de propriétés à travers la structure hiérarchique définie par la structure logique, il est nécessaire de définir/calculer l'ensemble des propriétés de présentation pour chaque type d'élément, même lorsque ce n'est pas significatif.

3. Les propriétés peuvent enfin être attachées à des *objets de formatage* (ou *formatting objects*) : par exemple des blocs de contenu imbriqués ou non, des colonnes, des pages. Ce niveau d'abstraction supplémentaire permet de s'affranchir des inconvénients précédents mais nécessite de spécifier comment les éléments du document sont transformés en objets de formatage. Le processus de formatage résultant est plus complexe à mettre en œuvre, notamment dans un contexte interactif dans lequel il est nécessaire de maintenir la correspondance entre les structures source (éléments de la structure logique) et les structures formatées. C'est l'approche choisie dans DSSSL et XSL.

Parmi ces trois méthodes d'affectation des propriétés de formatage, les deux premières correspondent à la production d'un arbre de structure spatiale qui est similaire à l'arbre de structure logique, tandis que la troisième méthode permet de construire un arbre de structure spatiale différent (les nœuds constituent les objets de formatage).

Dans la section 4.2.2 ci−dessous, je décris comment j'ai spécifié, dans le contexte d'un processus de formatage interactif, une approche intermédiaire entre la deuxième et la troisième méthode : l'objectif était de limiter la dépendance de la structure spatiale sur la structure logique sans pour autant mettre en œuvre le processus de transformation de la troisième approche qui est complexe et peu adapté à un environnement incrémental.

4.2.2 Le langage P et les modèles de pages pour les documents structurés

Comme je l'ai expliqué précédemment, le langage P de spécification de la présentation du système Thot est un langage de type déclaratif, qui appartient à la deuxième famille d'approches dans lesquelles les propriétés (appelées règles de présentation) sont attachées aux types d'éléments et sont regroupées dans des schémas de présentation [Quint 1994]. Par exemple dans la figure 5 ci−dessous, la règle *size: 12 pt;*

est attachée au type d'élément *livre*. Ces règles définissent la manière selon laquelle le contenu des boîtes attachées aux éléments doit être présenté (règles de style déclarées en absolu ou par héritage). Elles définissent également les paramètres de positionnement de ces boîtes les unes par rapport aux autres.

```
PRESENTATION Livre;
BOXES
  C1:                 { Modèle de colonne gauche }
      BEGIN
      Height: Enclosing . Height;
      Width: Enclosing . Width * 45%;
      HorizPos: Left = Enclosing . Left;
      END;
  C2:                 { Modèle de colonne droite }
      BEGIN
      Height: Enclosing . Height;
      Width: Enclosing . Width * 45%;
      HorizPos: Right = Enclosing . Right;
      END;
  Simple_Page :        { modèle de page }
      BEGIN
      Height: 25 cm;
      Width: 14 cm;
      Column (C1, C2);
      END;
RULES        { règles de présentation }
  livre :
      BEGIN
      Page(Simple_Page); { règle de mise en page }
      Size: 12 pt;
      END;
  auteur :
      BEGIN
      Size : Enclosing . Size + 2 pt;
      END;
  ...
END.
```

Figure 5 : Un exemple de modèle de structure spatiale pour un livre

Cependant, la spécification de modèles de pages est très limitée avec ce langage, notamment car, comme on l'a vu dans la section précédente, les propriétés de mise en pages demandent à être attachées à des objets de formatage et non directement à des types d'éléments de la structure logique. Ainsi, la structure formatée ne contient pas de

boîtes représentant les structures de page mais simplement des boîtes de séparation de pages attachées à des éléments de coupure de pages insérés dans l'arbre de structure logique.

Ma contribution dans ce domaine a consisté d'une part à étendre le modèle de pages du langage P, et d'autre part à proposer un mécanisme d'attachement des propriétés de structure spatiale à des structures de formatage qui sont directement issues de la structure logique.

Extension du modèle de pages et de colonnes pour le langage P

Ce modèle est caractérisé par une spécification en deux parties :

- La déclaration des boîtes de structure spatiale imbriquées les unes dans les autres selon le principe de la figure 4. Ainsi, dans l'exemple de la figure 5, les deux boîtes de colonnes *C1* et *C2* sont définies : leur hauteur (règle *Height*), leur largeur (règle *Width*) ainsi que leur position horizontale (règle *HorizPos*) sont définies relativement à leur boîte englobante (désignée par *Enclosing*). La règle *Column(C1, C2)* déclare l'inclusion des colonnes *C1* et *C2* dans la page de type *Simple_Page*.
- Les règles de mise en pages correspondent aux structures spatiales de plus haut niveau (les pages) et sont attachées aux types d'éléments, comme la règle *Page(Simple_Page)* pour l'élément *Livre* de la figure 5. Plusieurs règles de mise en pages peuvent être spécifiées dans un même schéma de présentation. Elles correspondent à un empilement des modèles de pages utilisés au fur et à mesure du formatage du document.

Structure de formatage

La structure de formatage initiale sur laquelle s'appuyait le système Thot est un arbre de boîtes directement issu de la structure logique à laquelle quelques nœuds sont ajoutés pour les coupures de pages. Il n'était pas envisageable d'appliquer ce même principe pour le formatage des documents en colonnes car les règles de dépendance induites par les imbrications de colonnes dans les pages n'auraient pu s'appuyer sur une telle structure (puisqu'elle ne contient que des délimiteurs de page). Aussi j'ai conçu une structure de formatage arborescente qui est un compromis entre la structure de formatage initiale de Thot et une structure de flots plus générale. Cette structure est caractérisée par deux parties (voir figure 6) :

- Les nœuds de structure spatiale sont placés en haut de l'arbre. Ainsi, les nœuds *HautDePage*, *CorpsDePage* et *BasDePage* définissent l'organisation graphique des pages et les nœuds *Col(C1)* et *Col(C2)* celle des colonnes.
- Les nœuds correspondant aux éléments de la structure logique sont placés comme descendants des nœuds de pages et de colonnes dans lesquelles ils sont contenus (le titre est sous la colonne 1 par exemple). Lorsqu'un élément s'étend sur plusieurs colonnes, il est nécessaire de lui attacher autant de nœuds que de zones dans lesquelles il s'affiche (le chapitre1 s'étend sur 2 colonnes sur la figure 6).

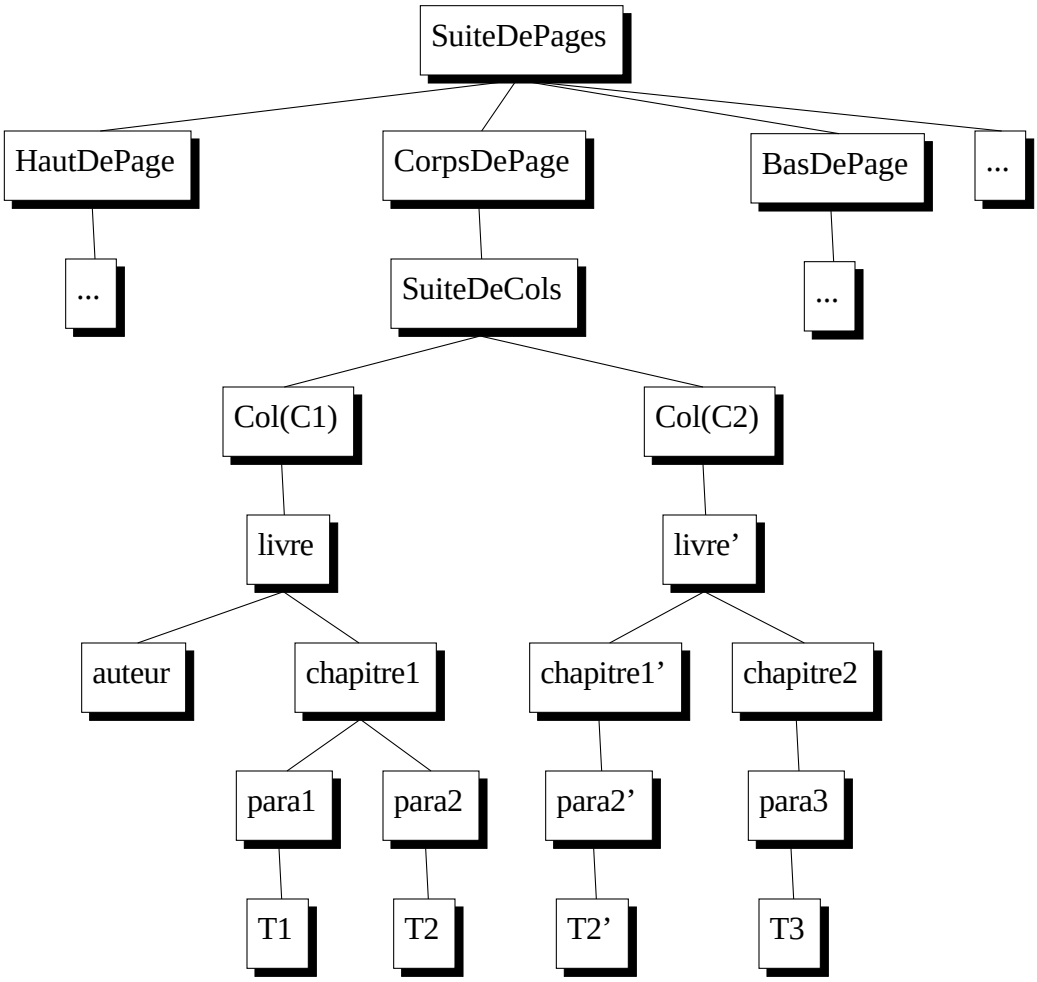


Figure 6 : Structure de formatage proposée pour Thot

Cette approche permet de maintenir une correspondance bi−directionnelle entre la structure logique et la structure spatiale, ce qui est indispensable dans un contexte d'édition interactive où l'utilisateur agit sur le document formaté pour modifier la structure logique. Par ailleurs, elle permet de profiter de façon optimale de la structure

logique pour porter les relations d'héritage de propriétés de style. Enfin, le processus de formatage peut utiliser de façon homogène tout l'arbre de structure spatiale car les relations d'inclusion spatiale sont respectées par tous les nœuds. Ce travail a été présenté lors de la conférence Electronic Publishing'94 [RV 1994].

4.2.3 Positionnement spatial de documents multimédia basé sur les contraintes

Le problème de la spécification de propriétés graphiques (en particulier du placement spatial) a suscité (et suscite encore) l'intérêt de nombreux chercheurs car il offre un bon cadre applicatif pour des techniques informatiques comme la programmation par contraintes ou les grammaires relationnelles [BLM 1997] [Weissman 1995]. On peut ramener la problématique du placement spatial à un problème de résolution de contraintes pour lequel la programmation par contraintes est une technique bien connue [Fron 1994].

Dans le projet Opéra, nous avons expérimenté cette technique pour le placement spatial d'objets multimédia dans l'environnement Madeus (cf. section 5.3). Ce travail a été effectué par Laurent Carcone dans le cadre de son stage d'ingénieur Cnam [Carcone 1997] et a été co−encadré par Muriel Jourdan et moi−même. Les premiers résultats ont été présentés lors du workshop LAMPE sur les modèles de présentation et les modèles de pages [CJR 1997]. Les principes sont résumés ci−dessous.

Comme dans les modèles précédents, cette approche est fondée sur un mode déclaratif du problème, par affectation de propriétés et de relations, qui offre une séparation claire entre la phase de définition et la phase de résolution. La principale différence réside dans le mode de résolution des relations : dans l'approche décrite précédemment, on cherche à tirer parti le plus possible de l'organisation logique du document y compris pour la définition du placement spatial : les relations sont principalement placées entre voisins dans la hiérarchie (englobant, descendant, précédent, suivant, etc.); avec une approche par contraintes, on ne s'appuie par sur la structure logique et les relations peuvent être placées entre n'importe quels éléments. Les relations spatiales spécifiées entre les objets produisent une structure de graphe de dépendances des variables de placement spatial. Cette approche de programmation par contraintes facilite la modélisation grâce à l'utilisation de langages souples et souvent proches des concepts de l'utilisateur comme *Video1 et Image1 alignées à gauche, Image2 au−dessus de Image1, Titre1 et Auteur1 centrés,....* Notons que dans cette approche, le modèle de structure spatiale est un graphe de boîtes (reflétant leurs dépendances) et non un arbre de boîtes (reflétant leur inclusion).

La spécification du placement spatial d'éléments de documents sous forme de relations est à la base des principes de construction des graphiques. C'est également un besoin dans le cadre de documents multimédia si l'on considère que les objets à placer sont plutôt à gros grain (une vidéo, une image, un message textuel, etc.) et si l'on cherche à rendre homogènes les modes de composition spatiale et temporelle. En effet, il y a beaucoup de similitudes entre le fait de vouloir placer un objet A à gauche d'un objet B et le fait de vouloir présenter A avant B.

Chaque objet du document est représenté par une boîte délimitée par le rectangle d'encombrement correspondant. Les paramètres qui la caractérisent sont *left*, *top*, *width* et *height* qui sont la position du coin supérieur gauche et ses dimensions. C'est l'ensemble de ces paramètres, pour tous les objets du document, qui constitue le groupe de variables sur lesquelles portent les contraintes de placement. Ces contraintes sont exprimées par des relations spatiales entre les objets qui définissent l'alignement, le centrage, le positionnement ou les dimensions des boîtes. Elles sont ensuite traduites en un ensemble d'équations linéaires où l'ordre de déclaration n'a pas d'importance. Le formatage final est alors réalisé par un résolveur de contraintes qui calcule les valeurs numériques des paramètres et définit ainsi la position de chaque objet à l'intérieur de l'objet composite qui le contient. Dans le contexte d'un environnement d'édition interactif, ce processus se ramène à un problème de maintien de solution d'un système de contraintes. L'auteur définit progressivement le placement qu'il désire obtenir en ajoutant ou retirant successivement des relations entre des éléments, ou bien en déplaçant ces derniers. Le résultat de chacune de ces actions doit prendre en compte le placement courant des éléments et les modifications apportées par cette action. Le résolveur de contraintes utilisé est *DeltaBlue*, qui fait partie d'une famille de résolveurs développés par l'université de Washington. C'est une technique de résolution de contraintes par *propagation locale* qui permet d'obtenir des performances adaptées à des contextes interactifs [SMFB 1993], [Freeman−Benson 1993], mais en contrepartie, induit certaines limitations comme l'impossibilité des gérer les cycles de contraintes et les inégalités.

Cette expérimentation a montré la pertinence de l'approche malgré ses limites et les deux versions du prototype Madeus (cf. section 5.3) s'appuient sur ce mode de spécification et sur cette technique de formatage spatial.

Les recherches en cours dans ce domaine considèrent soit des approches hybrides qui utilisent l'approche locale pour tous les cas pouvant être traités par cette approche, et l'approche globale pour les autres cas [BF 1995], [Trombettoni 1997], soit une

résolution globale avec un simplex incrémental qui assure des performances compatibles avec un contexte interactif [BLM 1997].

4.3 Modèles de représentation des structures temporelles

Les activités sur les structures temporelles des documents sont les plus récentes et les standards et outils pour spécifier et manipuler ces structures ne sont pas dans un état de stabilité et de complétude aussi avancé que pour les autres structures. Aussi, je décris dans cette partie le fruit de l'analyse que j'ai effectuée avec Muriel Jourdan et Nabil Layaïda sur les caractéristiques des documents multimédia [JLR 1998], [SLR 1999]. Je termine la section en présentant le modèle de documents multimédia Madeus (section 4.3.2) qui est au cœur des travaux menés dans le projet Opéra sur le thème du multimédia. L'utilisation de ce modèle est présenté en section 5.3.

4.3.1 Principes et caractéristiques des documents multimédia

L'introduction de la dimension temporelle dans les documents conduit nécessairement à de nouveaux besoins d'expression : la durée des objets, leur placement temporel et leur synchronisation. Ces informations temporelles doivent s'intégrer dans l'ensemble des informations attachées aux documents. Par exemple, pour spécifier le déplacement d'un objet sur l'écran, il est nécessaire d'exprimer une information spatiale pour la trajectoire, comme les positions initiale et finale correspondant au déplacement, ainsi qu'une information temporelle comme la durée du déplacement et sa synchronisation avec d'autres objets.

Le qualificatif de multimédia se rencontre dès lors qu'une application traite des documents qui ne sont plus seulement textuels (plusieurs média de base sont utilisés). C'est ainsi que des documents comportant uniquement du texte et des images sont appelés multimédia. À l'autre extrême de cette échelle, on trouve des documents caractérisés par une dynamique beaucoup plus importante grâce à l'intégration d'informations textuelles, d'images mais aussi du son, de la vidéo, des images animées, des éléments contrôlés par des programmes externes, etc.

Cependant, les documents multimédia sont non seulement caractérisés par des contenus de nature diverse mais aussi par l'organisation temporelle de leurs composants. Dans cette section, les unités d'information considérées comme atomiques sont appelées *objets multimédia* et la description de l'enchaînement des objets dans le temps est appelée *scénario temporel*.

4.3.1.1 Nature des objets multimédia

Les objets multimédia peuvent être classés en deux catégories de par leur mode de présentation dans le temps :

- *Les objets discrets* : leur contenu est délivré de façon instantanée, comme le texte et les images statiques.
- *Les objets continus* : leur contenu est délivré de façon progressive comme les vidéos, les sons ou les animations.

Ils peuvent aussi être caractérisés par leur mode de perception :

- *Les objets visibles*, c'est−à−dire que l'on peut afficher : texte, image, vidéo, animation.
- *Les objets audibles* comme les sons.

Pour chaque catégorie d'objets, différents formats de codage existent (Ascii pour les textes, jpeg, gif, tiff, png, pour les images, au, wave pour les audio, mpeg, ... pour les vidéos, etc.).

4.3.1.2 Style et activation

À ces différentes catégories d'objets, le système doit permettre d'attacher des informations de présentation ou *règles de style* (cf. section 4.2). Le jeu de règles dépend de chaque type de média, comme la fonte pour les textes ou le volume sonore pour les sons. Ces paramètres de présentation doivent pouvoir dépendre du temps de façon à réaliser des *effets de style* lors de la présentation d'un objet, comme par exemple, le déplacement d'un texte selon une trajectoire, la modification graduelle de la couleur, du volume du son, etc.

De même, on peut attacher un comportement d'activation à tout type d'objet. Un objet activable, appelé aussi *bouton*, est un objet sur lequel une action prédéfinie dans le document, par exemple l'activation d'un lien hypermédia, est déclenchée lorsqu'il est activé. Cela correspond à la définition d'une structure hypermédia pour le document (voir section 4.4). Là encore, cette propriété d'activation peut être définie pour un intervalle de temps durant lequel le lecteur aura la possibilité d'activer le lien.

4.3.1.3 Composition temporelle

La définition des structures temporelles d'un document consiste à spécifier les schémas de synchronisation entre les objets et les éléments composites du document.

Différents modes de composition temporelle ont été expérimentés, depuis la spécification par placement absolu ou *timeline* (spécification des dates de début et de fin des objets), l'utilisation de langages de programmation ou d'opérateurs de composition jusqu'à l'utilisation d'algèbres de relations. Nous avons fait un travail de synthèse sur ces différents modes de composition [JLR 1998], [SLR 1999] qui nous a permis de proposer une classification de ces techniques selon trois approches :

1. Les techniques opérationnelles ou impératives : axe de temps absolu (Director [Macromedia 1998], HyTime [ISO 1997]), programmation par scripts (Lingo dans Director, MHEG [Meyer−Boudnik 1995]).
2. Les techniques déclaratives par opérateurs temporels qui correspondent à la spécification par structures d'exécution à base d'arbres (CMIFed [RJMB 1993] et SMIL [W3C SMIL 1998]).
3. Les techniques déclaratives par relations temporelles, Firefly [BZ 1992], Isis [KS 95], Madeus [JLRST 1998], dans lesquelles l'auteur déclare les placements temporels souhaités sans donner toutes les informations temporelles attachées aux objets. La plupart de ces techniques s'appuient sur l'algèbre d'Allen [Allen 1983] pour la spécification des contraintes temporelles.

Il est intéressant de noter que tous les outils commerciaux actuels font appel à des *timelines* combinés à des approches impératives à base de scripts, se rapprochant plus d'outils de programmation que de véritables outils d'édition. Dans la suite de cette section, je précise le modèle de composition obtenu par une *approche à base de contraintes temporelles* puisque c'est celle qui est à la base de nos travaux.

Différents travaux théoriques, qui proviennent notamment du domaine de la planification en robotique dans lequel le problème de la synchronisation temporelle est crucial, ont permis de formaliser l'information temporelle des objets ainsi que des relations entre eux. Ainsi, tout objet multimédia peut être manipulé à travers trois informations temporelles principales (dont l'une est redondante) :

- Son instant de début.
- Sa durée de présentation.
- Son instant de fin.

Il existe ainsi deux façons de représenter le déroulement d'un scénario : à travers les changements qui surviennent, comme « *Lorsque la vidéo est terminée, le texte commence à s'afficher et la bande son démarre* », ou au contraire en reliant globalement

les activités entre elles comme « *le texte et la bande son démarrent en même temps et suivent la vidéo* ». Ceci débouche sur deux types de représentation :

- Une représentation fondée sur les *instants*, en particulier les instants de début et de fin des objets. Trois relations sont possibles : avant, après ou pendant.
- Une représentation fondée sur les *intervalles* correspondant à la durée des objets. Allen a étudié les relations possibles entre deux intervalles et a montré qu'elles se réduisaient à 13 relations, soit de nature parallèle (pendant, au_-démarrage, à_la_fin, en_même_temps, en_chevauchement), soit de nature séquentielle (avant, à_la_suite).

Cependant, d'autres opérateurs de composition sont nécessaires dans le contexte des présentations multimédia, notamment pour exprimer la *causalité*, comme lorsque l'occurrence d'un instant de fin d'un élément provoque l'arrêt d'un autre, même si ce dernier n'a pas restitué tout son contenu à l'utilisateur [DK 1995]. On parle alors de *relations d'interruption* (cf. section suivante).

4.3.2 Le modèle de documents de Madeus

Introduction

Le projet Opéra a débuté ses travaux sur les modèles temporels en 1994 dans le cadre de la thèse de Nabil Layaïda [Layaïda 1997] puis de celle de Loay Sabry−Ismail [Sabry 1999] que j'ai co−encadrées. Ce thème a rapidement pris de l'ampleur et nous sommes maintenant un groupe de 3 chercheurs et 2 doctorants sur le sujet.

Les différentes activités de recherche que nous menons sur le thème multimédia couvrent des aspects variés : modèles de représentation des informations temporelles et de leur intégration dans les autres caractéristiques des documents, principes de conception d'outils auteur et fonctions liées à l'exécution de ces documents. Ces différents travaux sont validés au travers de réalisations prototype qui sont intégrées dans une plate−forme commune appelée Madeus. L'objectif du projet Madeus est d'offrir une réponse globale au problème de l'édition et de la présentation de documents multimédia. J'ai commencé à prendre une part active à ces travaux lorsque j'ai pris en charge la gestion du projet à partir de mi−96 et, outre le co−encadrement des différentes thèses du projet, je me suis plus particulièrement impliquée dans les activités concernant l'intégration des modèles (logiques, spatiaux, temporels et de liens), l'analyse des besoins ainsi que dans la conception d'environnements auteur.

Dans cette section, je décris les principes qui sont à la base du modèle de documents de Madeus. Les travaux sur les outils auteur ainsi que sur le prototype Madeus sont présentés en section 5.3.

Madeus, un modèle à base de contraintes

Le modèle de Madeus [Layaïda 1997], [JRT 1998b] fait partie des modèles déclaratifs à base de contraintes. Comme on l'a vu, l'auteur spécifie ce qu'il souhaite obtenir comme placement temporel (par exemple, qu'une vidéo soit présentée après un texte et en même temps qu'un commentaire sonore) sans avoir à spécifier toutes les informations temporelles attachées à ces objets (instants de début et de fin, durée). Un scénario est donc défini par un ensemble de relations temporelles (avant, pendant, ...) entre les objets et une spécification de la durée souhaitée de chaque objet sous forme d'un intervalle de valeurs possibles.

Le jeu de relations temporelles est constitué des relations d'Allen (avec un paramètre de durée pour les relations avant, pendant, et de recouvrement) augmenté de relations d'interruption. Ainsi, la composition parallèle peut exprimer différentes sémantiques comme :

- La composition parallèle avec égalité de durée sans perte d'information des objets composés, c'est la sémantique de la relation *égale* d'Allen.
- La composition parallèle avec interruption, c'est−à−dire avec une perte éventuelle d'information d'un des composés :
  - soit lorsque le premier terminé provoque la terminaison du plus long ; cet opérateur causal est appelé *parmin*,
  - soit lorsque l'un des composés désigné explicitement se termine quel que soit l'état de l'autre (il est donc éventuellement coupé) ; cet opérateur est appelé *parmaster*.

À partir de cette liste de contraintes, le système peut vérifier la cohérence du document : l'existence d'au moins une solution (pour chaque objet, une date de début et une durée) qui respecte toutes les contraintes ; il peut ensuite produire statiquement et/ou dynamiquement une (des) solution(s) de placement temporel à partir des spécifications : par analogie aux systèmes d'édition conventionnels, on parle de *formatage temporel*. Le document formaté peut alors être présenté à l'auteur/ lecteur (cf. section 5.3).

Un document Madeus possède une *organisation logique* décrite sous forme d'une hiérarchie de composants, les feuilles étant des objets de base (texte, image.

vidéo, son). À chaque niveau de la hiérarchie, l'auteur peut spécifier des relations temporelles entre tous les objets (de base ou composites) qui partagent le même père. Ceci constitue le *scénario temporel* du document. La partie droite de la figure 7 est un exemple de scénario temporel spécifié selon cette syntaxe.

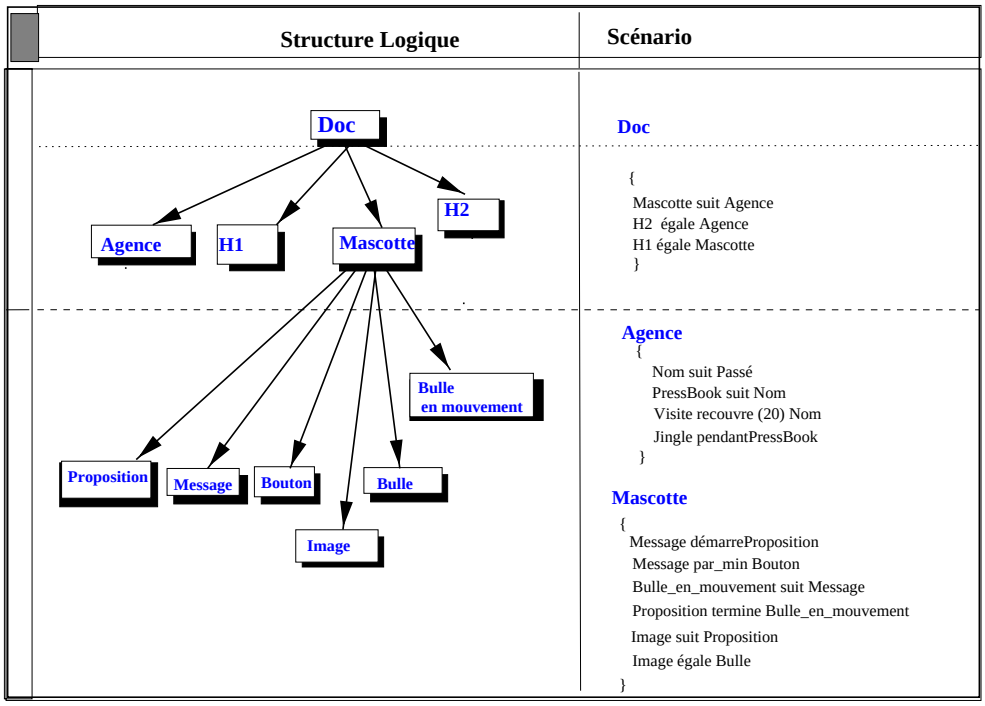


Figure 7 : Exemple de scénario temporel en Madeus

Bilan

Ce modèle a été utilisé pour spécifier de nombreux documents multimédia de démonstration. Une grande part de son intérêt repose d'une part sur son pouvoir d'expression, et d'autre part sur les possibilités qu'il offre dans un environnement auteur

(incrémentalité de la spécification, vérification de cohérence) ainsi que dans un environnement de présentation (capacité d'adaptation du scénario aux contraintes externes). Ces atouts dépendent de l'existence d'algorithmes qui gèrent le jeu de contraintes de ce langage. Si l'on sait vérifier et résoudre les scénarios exprimant des contraintes sur des objets contrôlables (auxquels on sait affecter une durée lors du formatage) selon le jeu de contraintes d'Allen, il n'existe pas encore de réponse satisfaisante pour des scénarios plus généraux dans lesquels la durée de certains objets n'est pas connue à l'avance. Nous avons cependant proposé dans [SLR 1999] des algorithmes qui permettent de compenser en partie ce type d'indéterminisme.

Enfin, ce modèle fait l'objet de nouveaux travaux (cf. section 6) qui visent à étendre son pouvoir d'expression dans deux directions : la spécification de relations spatio−temporelles (Dea de Lionel Villard [Villard 1998] que j'ai encadré) et la description structurée de vidéo permettant la composition de fragments vidéo (scènes, plans, etc.) avec d'autres objets (Dea de Tien Tran Thuong [Tran−Thuong 1999] que j'ai co−encadré). Ce dernier travail a fait l'objet d'une publication [RTV 1999].

4.4 Modèles de représentation des structures de navigation dans les documents

4.4.1 Introduction

Les liens hypertexte (ou hypermédia) permettent de décrire des informations sémantiques sous forme de relations entre éléments de documents qui ne sont pas les relations hiérarchiques. On peut représenter la structure sous−jacente d'un hypertexte par un graphe dont les nœuds sont les éléments de document et les arcs les liens. Les hypertextes et les hypermédia ont fait et font l'objet d'intenses travaux de recherche et le web est maintenant une illustration des plus concrètes de ce domaine [Nanard 1995]. Enfin, les standards existent comme HyTime [ISO 1997] et plus récemment, le langage XLink [W3C XLink 1999] en cours de spécification par le W3C.

Avec l'avènement du multimédia, de nouveaux besoins de liens sont à prendre en compte [Hardman 1998]. C'est sur ce sujet que nous avons travaillé dans le projet Opéra : nous avons proposé une typologie des modes de navigation dans les documents multimédia [SRL 1997] que nous avons mis en œuvre dans Madeus (voir section 5.3.3). Dans la section ci−dessous, je présente les principaux résultats de cette analyse de la navigation temporelle.

4.4.2 Principes de navigation temporelle

Les lecteurs cherchent à accéder aux informations d'un document imprimé de façons multiples : outre la lecture linéaire donnée par le parcours page après page du document, on peut citer l'accès depuis une table des matières, depuis des renvois comme les notes, l'accès au hasard, le parcours de la fin au début, etc. Dans le cadre d'une lecture électronique, cette fonction reste autant nécessaire et les environnements de navigation hypertexte ont démontré clairement l'intérêt du support électronique pour cette fonction.

Dans le cas des documents multimédia, la nature dynamique de l'information rend encore plus nécessaire et complexe la navigation : outre les hyperliens « classiques », étendus pour prendre en compte la dimension temporelle, des fonctions de parcours dans le temps sont aussi nécessaires, de façon similaire aux boutons d'accélération, pause, etc. des lecteurs de disques ou de cassettes audio ou vidéo. Nous distinguons donc :

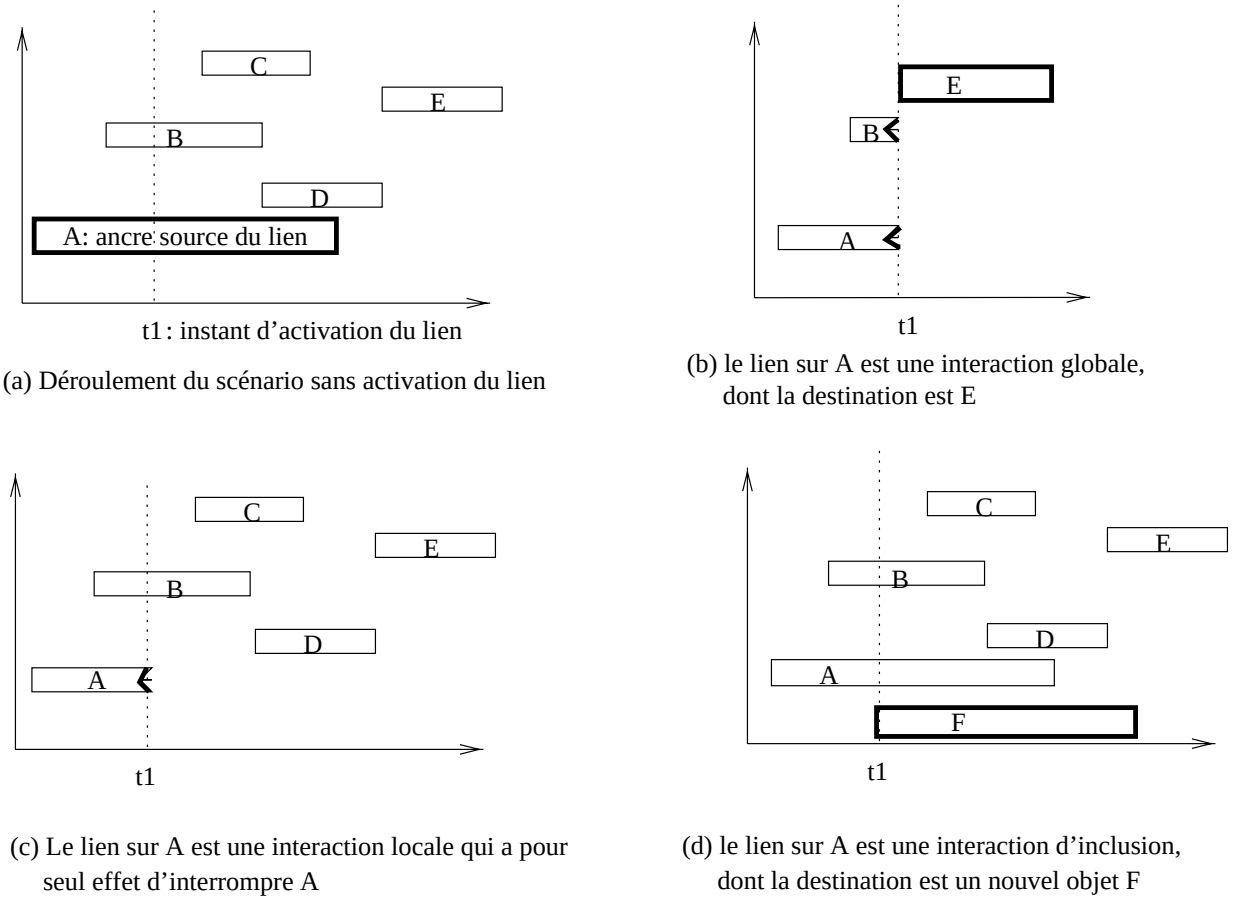
- *La navigation indépendante du document* : cette forme de navigation est fournie au travers de boutons de contrôle du temps (TAC : *Temporal Access Control*) au niveau du système de présentation, comme les boutons de pause, reprise, accélération en avant ou en arrière, saut d'une scène à l'autre, etc. Ces fonctions de navigation étant indépendantes du document présenté, elles n'apparaissent pas dans la spécification du scénario du document.
- *La navigation dépendante du document* : cette forme de navigation est conçue par l'auteur du document de façon à aider les lecteurs dans leur parcours du document. Elle fait donc partie de la spécification du document. Cette navigation est définie par des *objets activables*, comme les boutons d'interaction, que nous analysons ci−dessous.

Pour ce dernier type de navigation, il est possible d'identifier différentes interactions que les auteurs peuvent spécifier. Ces interactions peuvent être classées dans trois catégories en fonction de la sémantique temporelle qui leur est associée (la figure 8 illustre ces trois cas) :

- *Les interactions globales* (figure 8.b) : l'activation d'une interaction globale (comme le lien B sur la figure) a pour conséquence de stopper tous les objets en cours de présentation à cet instant pour démarrer la présentation d'un autre document (lien externe) ou d'une scène du document courant (lien intra−document : dans l'exemple la cible est définie comme l'instant de démarrage de l'objet E).

Un besoin associé à ce mode de navigation est le retour arrière comme dans les navigateurs HTML. Cependant, la dimension temporelle des documents multimédia introduit une plus grande difficulté pour la mise en œuvre de cette fonction puisqu'il faut revenir à l'instant de l'activation du lien, c'est−à−dire, être capable de continuer l'exécution de tous les objets à partir de l'instant où ils avaient été stoppés, au moment de l'activation du lien.

- *Les interactions locales* (figure 8.c) : l'activation d'une interaction locale a pour conséquence de ne stopper qu'une partie des objets en cours de présentation à cet instant (ici l'objet A lui−même). Ce type d'interaction est plus difficile à gérer car certaines synchronisations sont difficiles à assurer en présence d'interactions locales, par exemple si les objets interrompus par l'interaction sont en relation d'égalité avec des objets non interrompus.
- *Les interactions d'inclusion* (figure 8.d) : l'activation d'une interaction d'inclusion a pour conséquence d'ajouter un sous−scénario à partir de l'instant courant de la présentation, par exemple, un texte explicatif à une image affichée (l'objet F sur la figure). Les objets du sous−scénario inclus ne peuvent cependant pas être synchronisés avec les autres objets du document dans la mesure où ils sont insérés de façon asynchrone.



(a) Déroulement du scénario sans activation du lien

(b) le lien sur A est une interaction globale, dont la destination est E

(c) Le lien sur A est une interaction locale qui a pour seul effet d'interrompre A

(d) le lien sur A est une interaction d'inclusion, dont la destination est un nouvel objet F

Figure 8 : Les trois types d'interactions dépendantes du document

5 Applications documentaires

Avec l'utilisation de formats de documents de plus en plus riches, les applications de traitement de documents ne sont plus seulement limitées au seul domaine de la chaîne éditoriale (édition, publication, archivage). Il est en effet possible de tirer parti de l'information de ces formats (typage, lien, attributs) pour réaliser des applications comme la recherche d'information, la génération dynamique de documents à partir de bases documentaires, le couplage avec d'autres applications : le document devient alors un moyen de communication entre applications ou entre une application et un (des) utilisateur(s).

À l'inverse, il est possible d'identifier un certain nombre de fonctions de base liées aux documents qui sont nécessaires pour la mise en œuvre de ces applications : fonctions d'édition, formatage spatial, formatage temporel, fonctions de conversion, fonctions de restructuration, fonctions de recherche, fonctions de navigation, fonctions de coopération, ...

Dans ce chapitre, je décris les applications auxquelles j'ai participé dans le cadre du projet Opéra, à savoir :

- la transformation de documents structurés,
- les outils auteur de documents structurés,
- les environnements auteur pour documents multimédia.

Ces travaux s'appuient sur les modèles présentés dans la section précédente et en constituent une validation indispensable. Par ailleurs, ces différents outils nous ont permis d'établir de nombreuses coopérations scientifiques avec des partenaires académiques et industriels.

5.1 Transformation des documents structurés

Avec le développement du web, l'échange et la ré−utilisation de documents sont devenus une préoccupation de premier plan. Si XML offre une représentation unifiée des structures de document (cf. section 4.1.1), l'organisation et les types des éléments de ces structures sont toujours spécifiques aux applications qui les manipulent. Le besoin de transformer les documents pour pouvoir les partager entre applications est donc devenu une préoccupation de premier plan pour la conception d'applications XML.

Fondamentalement, le problème de la transformation de documents XML peut être posé de la façon suivante :

> *Étant donné un document appartenant à une classe, elle−même définie par une DTD, par quel procédé est−il possible de transformer l'intégralité ou une partie de ce document pour le rendre compatible avec une autre DTD ?*

En 1988 déjà, Furuta et Stotts [FS 1988] identifiaient le besoin de transformation pour la conception d'applications utilisant les documents structurés. Dans le projet Opéra, nous avons étudié ce problème depuis 1990 (thèse d'Extase Akpotsui). Outre les travaux sur la modélisation des types de documents que j'ai décrit en section 4.1.2, nous avons proposé et expérimenté des solutions selon différentes approches : transformations explicites, transformations automatiques et finalement nous avons proposé une approche hybride qui combine les deux précédentes (thèse de Stéphane Bonhomme que j'ai co−encadrée). Après avoir situé les besoins de transformation dans la chaîne documentaire, je décris les principes de transformation que nous avons proposés.

5.1.1 Étude des besoins

On peut identifier les transformations nécessaires non seulement pour l'édition de documents structurés, mais pour tout le cycle de vie de ces documents :

- Importation de documents pour permettre la réutilisation des documents existants.

  Les documents formatés utilisent des formats aux syntaxes trop diverses pour pouvoir être importés selon une méthode générique. Ils sont importés à l'aide de filtres spécialisés qui produisent des documents XML conformes à des DTD d'acquisition. Une transformation supplémentaire est nécessaire pour convertir ces documents vers une DTD correspondant au besoin d'une application particulière.
- Transformation en cours d'édition : les transformations intervenant au cours de l'édition permettent la modification dynamique de la structure du document. Ces transformations incluent le changement de type d'éléments, l'insertion et la suppression d'éléments de structure intermédiaires et le déplacement d'éléments dans la structure du document.
- Conversion de documents entre modèles : cette classe de transformations permet de convertir un document dans son intégralité d'une DTD d'édition dans une autre. Ces transformations sont nécessaires lorsqu'une DTD évolue ou lorsque les documents doivent être échangés entre applications utilisant des

DTD différentes. Par exemple lors de la sauvegarde de documents, il est parfois nécessaire de les convertir dans un format spécifique pour publication dans une revue, ou en HTML pour publication sur le web.

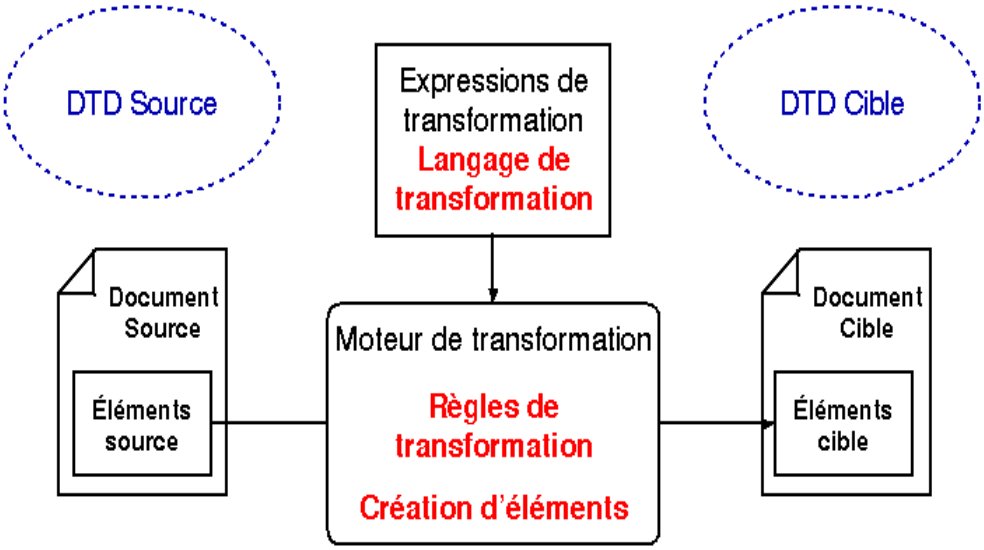


Figure 9 : Transformation explicite de document

5.1.2 Techniques de transformation

Pour répondre aux besoins de transformation ci−dessus, trois approches ont été expérimentées dans le projet Opéra :

La transformation explicite

La transformation explicite se fonde sur un ensemble de règles guidant la transformation (voir figure 9). Ces transformations par règles ont un grand avantage sur les filtres qui ne traitent que des documents appartenant à un unique format ou à une même DTD source et qui produisent un document d'un format préalablement défini. Les règles permettent de paramétrer les transformations qui peuvent ainsi s'appliquer à différents formats de documents. Cette technique peut être mise en œuvre par des applications dédiées aux transformations ou intégrées dans d'autres applications (éditeurs de documents, bases de données). Les langages de transformation utilisés dans ces systèmes peuvent être de nature déclarative, comme les langages d'IDM [Digithome 1996], de Chameleon [MKNS 1989] ou de XSL [W3C XSL 1999], impérative (Balise [Balise 1996], CoST 2 [English 1998]) ou par requêtes (SgmlQL [LMR 1998], Scrimshaw [AR 1997]).

Nous avons expérimenté cette technique de transformation explicite dans un environnement d'édition de documents HTML, l'éditeur Amaya du W3C. L'intégration de ce système dans un éditeur interactif lui confère quelques spécificités que n'ont pas les systèmes de transformation cités précédemment : processus en deux étapes pour donner à l'utilisateur le choix de la transformation à effectuer parmi les requêtes valides, représentation mémoire des requêtes sous forme d'automates pour faciliter la reconnaissance des requêtes valides lors de chaque demande de transformation d'une instance source. Ce système de transformation est présenté dans [BR 1996] et dans [Bonhomme 1998].

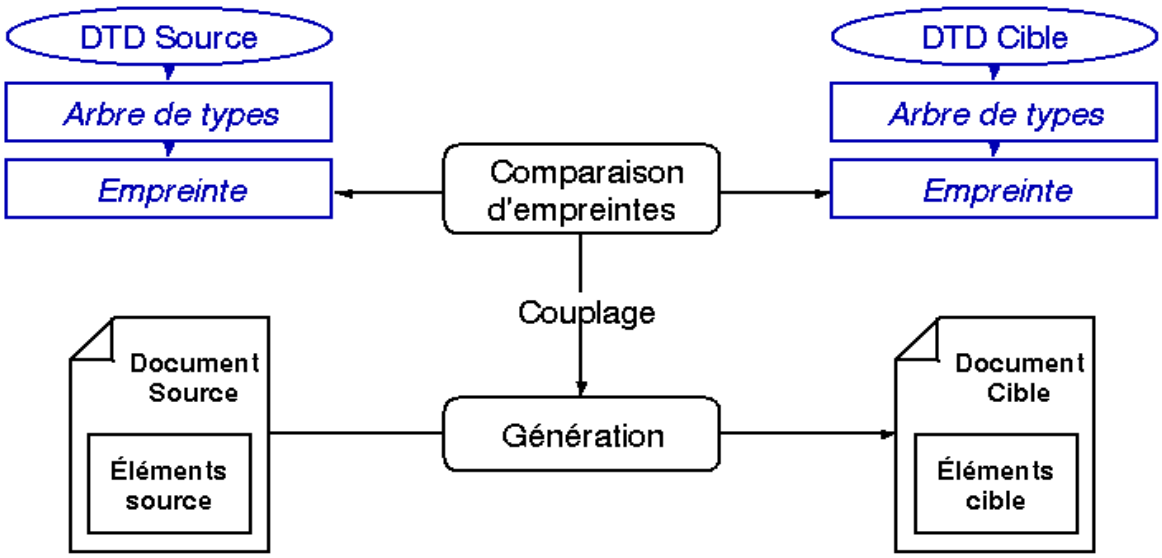


Figure 10 : Transformation automatique de document

La transformation automatique

La comparaison de DTD permet une **transformation automatique** de documents selon le principe de la figure 10 proposé dans [AQR 1992]. Pour effectuer la comparaison entre une DTD source et une DTD cible, l'ensemble des règles constituant chaque DTD est représenté sous forme d'un arbre qui est linéarisé avant comparaison (cf. section 4.1.2). Le résultat d'une comparaison est un ensemble de relations appelé *couplage* entre un type source et un type cible. La génération d'un document transformé peut alors être effectuée à l'aide de ce couplage. Cette approche est décrite dans [RCA 1997] et [BR 1997].

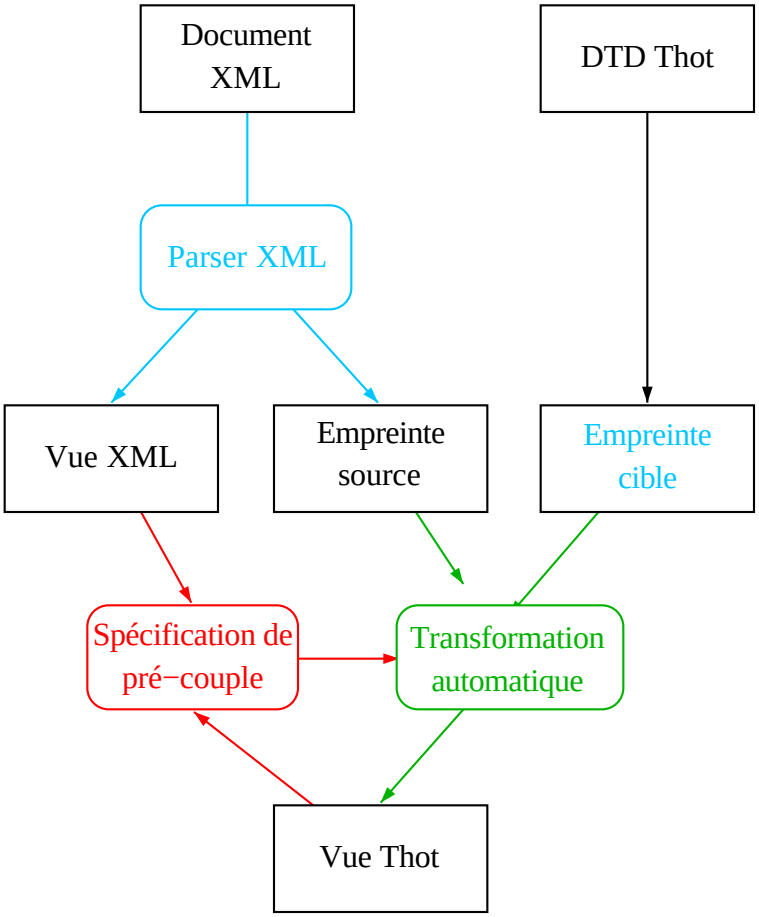


Figure 11 : Processus de conversion de documents XML en documents Thot

La transformation hybride

La transformation hybride s'appuie sur les résultats de l'expérimentation des techniques explicite et automatique. Cette approche originale, décrite dans [Bonhomme 1998] et [BR 1999], tire parti des avantages des deux techniques précédentes :

- la partie automatique évite à l'utilisateur d'expliciter toutes les transformations nécessaires ;
- et la partie explicite (spécification de pré−couples, cf. figure 11) permet à l'utilisateur de mieux contrôler le résultat des transformations obtenues.

Pour combiner ces techniques, deux méthodes sont envisageables : soit intégrer des transformations automatiques pour des sous−parties identifiées dans un processus de transformation explicite, soit paramétrer un algorithme de transformation automatique par des expressions données par l'utilisateur. C'est cette dernière solution que nous avons développée car elle est adaptée à un processus incrémental (comme dans le

cas de l'application de conversion interactive de documents XML en documents Thot dont la figure 11 donne le principe).

5.1.3 Bilan

Ces différents travaux ont abouti à des résultats intéressants, tant sur le plan pratique (les transformations dans l'éditeur Amaya apportent effectivement un meilleur confort à l'édition) que sur le plan théorique (les algorithmes développés pour la comparaison et pour la transformation automatique apportent une réponse aux besoins de transformation de structures XML dans les applications). C'est un domaine qui va prendre de l'ampleur du fait du développement des applications XML.

Ces travaux se poursuivent actuellement dans le projet Opéra dans le cadre du programme Génie (avec Dassault Aviation et Aérospatiale).

5.2 Outils auteur de documents structurés

Les environnements d'édition réalisés dans le projet Opéra ont permis de valider les choix de modélisation des documents structurés et de proposer des solutions à la mise en œuvre d'outils « wysiwyg » pour ces types de documents. Ainsi, pour réaliser les fonctions d'édition, de formatage et de coopération de ces outils, on cherche à tirer parti des modèles génériques utilisés pour spécifier la structure logique et la présentation graphique des documents.

5.2.1 Thot

Le cœur des différents environnements réalisés dans le projet Opéra est l'éditeur ThotEditor [Quint 1994], [Opéra 1997]. ThotEditor permet de créer, de modifier et de consulter de façon interactive des documents qui respectent des modèles (cf. section 4.1). Grâce à ces modèles, on obtient des documents homogènes et l'utilisateur peut se concentrer sur l'organisation et le contenu des documents qu'il traite, sans s'occuper de formatage ou de typographie, ces fonctions étant prises en charge par le système. ThotEditor effectue également d'autres traitements pour l'utilisateur, comme les numérotations, le maintien des références croisées, la gestion des index, la correction orthographique, etc. ThotEditor permet la production de documents dans de nombreux formats comme Postscript, Latex, HTML ou XML.

Une grande partie des fonctions de cet éditeur est accessible à travers une boîte à outils et un générateur d'applications (appelés Thot). Ainsi, d'autres applications peuvent construire leur interface et y attacher des traitements (actions) qui leur sont

propres, ou qui sont des fonctions standard du noyau d'édition Thot. Grâce au modèle de document structuré, il s'agit de fonctions de haut niveau qui prennent en charge les tâches les plus complexes pour les applications interactives. Les applications qui traitent les documents manipulent seulement leur structure et leur contenu ; l'affichage et le (re)formatage sont effectués de façon incrémentale par les fonctions de la boîte à outils Thot [RV 1994].

La librairie d'édition Thot a été diffusée depuis plusieurs années dans le cadre de contrats de collaboration (une dizaine). Depuis février 1997 cette diffusion se fait par ftp anonyme sous forme de code source. L'éditeur de documents structurés ThotEditor était déjà diffusé sous forme binaire depuis janvier 1996. Ces diffusions s'accompagnent de nombreuses documentations : celle de l'éditeur [BQRRV 1997] ainsi que celles de la boîte à outils et du générateur d'applications. Toutes sont accessibles depuis le serveur web du projet [Opéra 1997].

5.2.2 Ma participation à Thot

Ma participation à ces travaux a concerné principalement deux parties du logiciel Thot : d'une part l'expérimentation des principes présentés en section 4.2.2 pour le formatage complexe de pages et de colonnes, et d'autre part une refonte du programme d'impression de Thot pour permettre le formatage des pages au fur et à mesure de l'impression du document.

J'ai également contribué à la rédaction du manuel utilisateur ainsi qu'aux tâches liées à la diffusion de ce logiciel (gestion de la liste de diffusion, correction des erreurs, etc.).

5.2.3 Travaux construits à partir de Thot

De nombreuses expérimentations se sont greffées autour de cette boîte à outils, avec des objectifs variés : édition coopérative, édition pour le web ou réalisation d'un environnement hypertexte pour les généticiens des textes :

- Alliance [DR 1996] : c'est un environnement d'édition coopérative de documents structurés qui utilise la structuration logique des documents pour réaliser les services de coopération. Cet environnement s'appuie sur les fonctions d'édition de Thot qui sont complétées par des services spécifiques de coopération assurant un contrôle de cohérence de la coopération *a priori*, c'est−à−dire ne nécessitant pas de mécanisme de fusion de versions (voir thèse de Manuel Romero Salcedo [Romero 1998]). Je n'ai pas directement

participé à ces travaux sauf en ce qui concerne la gestion de son expérimentation dans le cadre du projet Storia avec ST Microelectronics [SR 1997].

- Amaya : c'est un client auteur et navigateur pour documents du web qui est réalisé par le W3C [QV 1997]. Les premières versions de cet outil ont été réalisées dans le cadre du projet Opéra [QRV 1995].
- Byzance : c'est un prototype qui vise à répondre aux besoins d'édition coopérative de documents du web. Sa première version a été développée dans le cadre de la deuxième phase du projet Storia [CR 1998]. Ce logiciel s'appuie sur Amaya pour les fonctions d'édition et de navigation, et sur Alliance pour les fonctions de coopération. De plus, un ensemble de fonctions spécifiques a été réalisé pour prendre en compte le contexte particulier induit par ce mode de coopération : l'espace de coopération étant différent de celui de l'espace de publication sur le serveur web, il est nécessaire d'offrir à l'utilisateur une fonction de publication différente de celle de sauvegarde dans l'espace de coopération. En complément, une politique de désignation (à l'ouverture, lors de la création de liens et lors de la publication) a été mise en œuvre de façon que les liens entre documents soient cohérents dans les deux espaces. Ce travail, dont j'assure la coordination, est maintenant effectué par Laurent Carcone dans le cadre du programme Génie avec les sociétés Aérospatiale et Dassault Aviation.
- Environnement pour la critique génétique des textes : c'est un travail mené par Jacques André et Hélene Richy à l'Irisa de Rennes dont l'objectif [AR 1997] est de montrer comment les documents structurés peuvent contribuer à l'informatisation des postes de travail pour les généticiens des textes grâce notamment à la structuration, au formatage multivues et aux liens hypertexte typés.

5.2.4 Bilan

Le savoir−faire acquis par le projet Opéra dans le domaine de l'édition de documents structurés lui a permis d'établir de nombreuses coopérations scientifiques et industrielles, tout d'abord avec la société Grif S.A., vers laquelle un premier transfert industriel a été effectué à partir de 1988. Depuis mon arrivée dans le projet Opéra, et surtout depuis que j'en assure la gestion, j'ai été amenée à participer et à assurer le suivi de ces collaborations : avec Dassault Aviation et Aérospatiale dans le cadre du programme Génie du Mensr et avec ST Microelectronics dans le cadre du projet Storia

[Inria Rhône−Alpes 1998]. Les collaborations scientifiques ont pris la forme d'expérimentations développées à partir de la boîte à outils, comme avec l'EPFL (équipe de Giovanni Coray et Christine Vanoirbeek).

5.3 Environnements auteur de documents multimédia

Comme décrit en section 4.3, les activités sur les documents multimédia constituent maintenant un des axes majeurs du projet Opéra. Outre les travaux sur la modélisation de l'information temporelle (cf.section 4.3.1), nous portons notre effort dans deux directions qui concernent les applications multimédia :

- L'analyse des besoins des auteurs de documents multimédia (voir section 5.3.1) qui nous a amenés à proposer des mécanismes de base pour la conception d'environnements d'édition multimédia (voir section 5.3.2).
- L'étude des fonctions de présentation de documents multimédia et de leur mise en œuvre dans les environnements auteur. Ces travaux sont décrits en section 5.3.3.

Le premier axe est principalement mené par Muriel Jourdan et moi−même (thèse de Laurent Tardif que nous co−encadrons) tandis que le second est pris en charge par Nabil Layaïda et moi−même (thèse de Loay Sabry [Sabry 1999]).

Dans les sections suivantes, je présente tout d'abord les résultats de notre analyse des besoins dans l'édition multimédia, puis je décris les principes de conception d'environnements auteur que nous proposons, et enfin les fonctions nécessaires pour la présentation des documents multimédia.

5.3.1 Les problèmes posés par l'édition de documents multimédia

Dans les systèmes traditionnels de traitement de documents orientés texte, le mode de communication entre l'utilisateur et l'application est caractérisé par :

- la présentation à l'écran d'une information principalement spatiale et statique,
- et la lecture active de l'utilisateur qui peut naviguer à travers l'information affichée.

Cette répartition des rôles « actif−passif » est inversée dans les systèmes de communication de l'audio et de la vidéo : l'information est de nature dynamique tandis que le lecteur/auditeur est passif. Dans le contexte des documents multimédia interactifs, ces deux rôles doivent être combinés dans le système de présentation puisque

l'information manipulée est à la fois de nature spatiale et temporelle. De plus, le système doit permettre certaines formes d'interaction sur le document présenté, comme la navigation intra− ou extra−documents par le biais d'hyperliens.

La nature dynamique des objets manipulés, tels que la vidéo et l'audio, ainsi que la définition de leur ordonnancement temporel (le scénario) rendent plus complexe la réalisation d'outils auteur. Le principe statique du wysiwyg, dans lequel l'information présentée à tout instant du processus d'édition correspond à l'information finale, ne peut s'appliquer à l'édition du scénario temporel des documents multimédia. Il n'est en effet pas possible de spécifier un comportement dynamique, par exemple un enchaînement entre deux vidéos, et d'en percevoir de façon immédiate et instantanée le résultat. C'est pourquoi nous distinguons deux étapes dans le processus de conception des documents multimédia : l'*étape de spécification* (ou édition) et l'*étape de présentation* (ou exécution).

À partir de l'analyse des possibilités d'expression des langages de spécification des scénarios temporels (cf section 4.3.1), nous avons identifié un ensemble de fonctions de base qu'un environnement de composition multimédia doit fournir aux auteurs [JLR 1998], [JLR 1999] :

- Le langage de spécification doit être utilisable sans compétence en programmation informatique.
- Le langage doit permettre le passage simple et souple de la représentation mentale que se fait l'auteur du document à un véritable scénario formalisé dans ce langage.
- Le langage de spécification doit être modulaire pour permettre à l'auteur de réduire la complexité de son document et aussi de réutiliser des parties déjà spécifiées.
- L'environnement auteur doit permettre de modifier facilement les documents en cours de conception de façon à permettre un processus incrémental de création.
- L'environnement auteur doit offrir des supports visuels permettant de percevoir le scénario en cours de spécification.

5.3.2 Environnement auteur de documents multimédia

Les activités décrites ici sont le fruit d'un travail effectué par de nombreuses personnes du projet (stagiaires, ingénieurs, doctorants et chercheurs). En effet, l'objectif général étant la conception d'environnements auteur qui répondent aux différents

critères identifiés dans la section précédente, il est clair que les résultats n'ont de sens que s'ils sont validés par la réalisation de prototypes qui nécessitent une force de travail importante et dont j'ai assuré une part de l'encadrement.

Trois mots−clés résument les principes de conception d'environnements auteur de documents multimédia que nous proposons : multivues, contraintes et extension. Plus précisément, les axes de travail de ce thème de recherche sont :

- La conception d'un environnement multivues pour l'édition de documents multimédia [JRT 1998c]. L'idée de base est que les multiples facettes d'un document multimédia sont plus facilement accessibles à l'auteur au travers de différentes vues qui sont autant de filtres sur le document. Ainsi, outre la *vue de présentation* qui permet de jouer le document, on peut définir une *vue des objets* organisée sous forme d'une hiérarchie de composants ainsi qu'une *vue du scénario* qui visualise l'enchaînement temporel du document (Dea de Laurent Tardif [Tardif 1997], présenté également dans [JRT 1997] et [JRT 1998a]). Ces vues peuvent être synchronisées de telle sorte que lorsqu'un objet est sélectionné dans l'une d'elles, les autres vues sélectionnent également cet objet en modifiant en conséquence leur affichage. Enfin, les opérations d'édition peuvent être effectuées dans chacune de ces vues.
- L'étude et l'utilisation de la technologie des contraintes pour la conception d'environnements auteur multimédia [LAY 1997], [JRT 1998b]. Les contraintes sont considérées pour répondre à deux types de besoins : (1) pour offrir un mode de spécification déclaratif sous la forme de définitions de propriétés partielles, relatives et non absolues (cf.section 4.3.2) ; (2) pour permettre d'offrir un support incrémental à l'édition par la vérification statique et incrémentale de la cohérence des scénarios.
- La conception d'une boîte à outils d'édition indépendante du langage source [JRT 1999b]. Cette boîte à outils permet d'une part d'utiliser les résultats précédents pour d'autres langages que Madeus, et d'autre part de les étendre pour prendre en compte les besoins spécifiques de ces langages : ajout de nouvelles vues ou de nouveaux services d'édition.

Les points forts des environnements auteur conçus selon ces principes sont d'une part les possibilités liées à l'usage de contraintes (vérification du scénario, incrémentalité) et d'autre part l'intégration de fonctions de présentation au sein de l'environnement d'édition.

Expérimentations

Les expérimentations faites dans le cadre de la thèse de Nabil Layaïda ont principalement porté sur la mise en œuvre de Madeus, un environnement d'édition et de présentation de documents multimédia à partir d'un modèle à base de contraintes dont les fonctions d'édition restaient limitées. Nous avons ensuite travaillé à la conception d'un environnement auteur qui réponde mieux aux objectifs identifiés dans la section précédente. Dans un premier temps, nous avons apporté des extensions au prototype Madeus [JLRST 1998], puis nous avons spécifié et redéveloppé l'ensemble du prototype selon une approche par objets (dans le cadre d'un contrat industriel avec la société Alcatel [JRT 1998d], [CJRTV 1999]). C'est à l'occasion de cette refonte que nous avons cherché à identifier les composants du logiciel indépendants du format source pour les offrir sous forme d'une boîte à outils appelée *Mikado puis Kaomi* [JRT 1999b]. Cette boîte à outils est utilisée en interne pour le développement des environnements d'édition *Smil−Éditeur* [JRTV 1999] (pour le langage Smil) et *Madeus*. Elle est également expérimentée en externe par Aérospatiale pour la réalisation d'un prototype de gestion d'un fonds documentaire pour la documentation technique aéronautique [DRTV 1999]. Enfin, nous sommes en train de réaliser un environnement auteur pour le langage MHML (format défini par Alcatel pour la description en XML du langage MHEG [Meyer−Boudnik 1995]) dans le cadre d'un contrat industriel.

5.3.3 Présentation de documents multimédia

La présentation (ou exécution) d'un document multimédia est nécessaire lors de différentes phases de son cycle de vie : lors de sa création et lors de sa consultation. Elle consiste en la restitution des objets média qu'il contient en respectant d'une part les contraintes internes représentées par les relations de composition entre ces objets et d'autre part les contraintes externes imposées par la disponibilité des ressources de la plate−forme d'exécution [LRS99]. La prise en compte de ces contraintes soulève de nombreux problèmes dans la réalisation des fonctions de présentation comme la synchronisation, la navigation temporelle et la prise en compte du comportement indéterministe de certains objets :

- La synchronisation de la présentation comprend la *synchronisation intra−objets*, c'est−à−dire entre les composants d'un même objet média (par exemple pour assurer la succession régulière des images d'une vidéo ou des échantillons d'une audio) et la *synchronisation inter−objets* entre des objets

différents : c'est celle qui est spécifiée par l'auteur dans le scénario temporel du document (par exemple, un texte s'affiche en même temps qu'une vidéo commence).

- La navigation temporelle consiste à offrir les différents modes de parcours dans le temps que nous avons identifiés en section 4.4.2.
- La gestion de l'indéterminisme consiste à trouver *a priori* ou *a posteriori* des stratégies qui permettent au système de présentation de réagir lorsque certains objets média ne respectent pas leur durée de présentation prévue (on parle alors d'*objets incontrôlables*).

Ces différentes fonctions ont été étudiées par Loay Sabry dans sa thèse [Sabry 1999]. Il a proposé des solutions qu'il a expérimentées dans le prototype Madeus et dont le résultat est une machine de présentation multimédia de type prédictif−réactif. Cette machine s'appuie en effet sur un hypergraphe temporel, résultat du formatage temporel du scénario qui offre le principal support pour des opérations de présentation (lancement / arrêt des objets média). Les événements asynchrones (navigation, désynchronisation) sont pris en compte par des algorithmes qui cherchent à remettre l'exécution le plus rapidement possible en cohérence avec le scénario. Je décris ci−dessous les deux tâches auxquelles j'ai participé le plus activement : la navigation temporelle et la gestion de l'indéterminisme.

Navigation temporelle

Dans Madeus, nous avons proposé un nouveau mode de parcours des documents multimédia à travers leur déroulement temporel [SRL 1997]. En s'appuyant sur la structure logique (composites) et la structure temporelle (synchronisation des débuts et fins d'objets), nous avons spécifié une navigation à la fois structurale et temporelle : il est possible de dérouler pas à pas un document en accédant directement aux instants significatifs de sa présentation : ces instants peuvent être choisis à partir de la structure logique (par exemple, le début de tous les composites d'un niveau donné de la hiérarchie) ou de la structure temporelle (d'un début/fin d'objet à l'autre). La réalisation de ce mode de navigation implique d'être capable de calculer l'état d'un document pour tout instant, c'est−à−dire la liste des objets actifs avec leur décalage temporel par rapport à leur instant de début : c'est ce qu'on appelle le *contexte de présentation*. En fonction des actions effectuées par l'utilisateur, les contextes sont calculés, empilés ou dépilés.

Prise en compte de l'indéterminisme dans les présentations

Deux approches complémentaires ont été spécifiées et expérimentées pour permettre d'adapter l'exécution d'un document multimédia soumis à des contraintes externes [SLR 1999]. Ces approches tirent parti de la structure de l'hypergraphe ainsi que de la flexibilité de certains objets (par exemple, la durée de présentation d'une image ou d'un texte peut être augmentée ou diminuée sans conséquence grave pour le lecteur).

Il s'agit dans la première approche de reformater dynamiquement le document en modifiant la durée de certains objets tout en respectant les spécifications initiales de l'auteur. Dans [LS 1996a], il a été montré que pour un ensemble de situations simples, il était possible de vérifier statiquement la validité de ces modifications.

Lorsque le formatage dynamique ne suffit pas pour rétablir une exécution conforme aux synchronisations spécifiées, la deuxième approche est mise en œuvre : un algorithme de réparation est exécuté pour limiter la durée pendant laquelle les désynchronisations persistent et pour réduire l'importance de ces désynchronisations. Le principe est de décaler le début ou la fin de certains objets en insérant dynamiquement des délais.

Les expériences effectuées dans Madeus ont permis de montrer que ces algorithmes réduisaient de plus de 50% les durées de désynchronisation, y compris dans des situations où plusieurs objets incontrôlables étaient joués.

5.3.4 Bilan

Les différents travaux qui sont menés sur le thème des documents multimédia dans notre projet illustrent bien le fait que c'est un domaine qui touche à de nombreux secteurs scientifiques : les contraintes, les interfaces, le système et les architectures logicielles.

Une autre caractéristique importante de nos travaux est le choix d'intégrer les résultats, notamment par le biais des prototypes réalisés. Ainsi, le couplage de l'environnement auteur avec la machine de présentation apporte une réponse aux besoins de perception rapide qu'ont les auteurs. Ce couplage induit également de nouveaux problèmes intéressants dans la conception de l'environnement auteur. Par exemple, quelle information faut−il présenter dans la vue scénario : le placement temporel effectif résultant d'une exécution ? le placement temporel résultat du formatage ? ou encore l'ensemble des solutions définies par le scénario ?

6 Orientations de recherche dans le domaine des documents

Après un rapide tour d'horizon sur les thèmes émergents qui touchent aux documents électroniques, je présente quelques pistes que nous considérons actuellement dans le projet Opéra.

6.1 Perspectives générales

Il est toujours délicat de proposer des directions futures d'un travail de recherche, et en particulier dans celui qui m'intéresse : le concept de document électronique a tellement évolué au cours des dernières années ! Parmi les nouvelles caractéristiques à considérer maintenant, on peut citer : la dynamicité du contenu (génération au vol), la temporalité, la réactivité, la mobilité, l'accessibilité, le traitement multi−utilisateurs et multi−applications.

Les évolutions actuelles amènent à aborder les problèmes non pas seulement au travers de la notion de document structuré et multimédia (structure d'information relativement statique) mais aussi sous l'angle de la notion d'application multimédia (structure d'information avec son comportement).

Un autre facteur important dont il est encore difficile de mesurer l'impact est celui du boum lié à XML : les résultats scientifiques établis depuis de nombreuses années sont (re)découverts par les entreprises. Et même, les techniques documentaires se trouvent de plus en plus utilisées dans d'autres domaines que celui du traitement des documents (XML−base de données, XML−commerce, XML−connaissance). Les documents ou plus généralement les *informations structurées* sont au cœur de nombreuses applications !

6.2 Axes de travail du projet Opéra

Depuis 2 ans, le projet Opéra s'était fortement orienté vers la problématique de la modélisation et du traitement du temps dans les documents, puisque c'était un manque important qui soulevait des problèmes à la fois théoriques et pratiques. L'évolution actuelle va vers une synthèse des différentes composantes du projet (documents structurés, coopératif, multimédia) pour aborder de façon intégrée les nouveaux sujets de recherche issus de l'évolution de ce domaine, notamment des documents dans le contexte réseau. En effet, le web donne une autre dimension aux problèmes abordés :

facteur d'échelle, dynamicité, indéterminisme, répartition. Plus précisément, nous considérons les axes suivants :

- Les modèles génériques de documents qui prennent en compte tous les aspects des documents, notamment le temporel et l'interactivité.
- Les extensions de ces modèles pour permettre la représentation structurée de média, comme la vidéo ou le mouvement d'objets, qui sont traités jusqu'à présent de façon non décomposable (sous forme d'un flot continu par exemple).
- Les outils auteur qui permettent aux concepteurs d'applications de maîtriser la complexité de ces documents dans toutes leurs caractéristiques: structuration, temps, partage entre créateurs, capacité d'adaptation des documents produits, etc.
- Les stratégies d'ordonnancement qui prennent en compte l'indéterminisme dû à la nature et au mode d'accès distant des composants d'un document multimédia.
- Les protocoles et architectures logicielles adaptés à ce type d'application : protocoles de coopération, protocoles de transmission temps réel permettant de gérer la répartition des informations à travers le réseau.

Je détaille ci−dessous les trois premiers points dans lesquels je suis plus particulièrement impliquée actuellement.

6.3 Modèles génériques de documents multimédia

Jusqu'à présent, les travaux dans le domaine de la spécification des documents multimédia ont majoritairement abordé le problème en considérant la spécification d'un unique document selon une approche dite « spécifique ».

Pourtant, de nouveaux besoins commencent à émerger avec l'utilisation grandissante de l'information multimédia dans des domaines d'application aussi variés que l'éducation, la médecine ou le tourisme. Pour limiter les coûts de réalisation en factorisant au mieux les tâches de conception de documents multimédia, il est nécessaire d'offrir le moyen de spécifier les documents à un niveau générique à travers la notion de *classe de documents*.

Un nouvel axe de recherche émerge donc : « la spécification de modèles génériques de documents multimédia ». On a vu comment les standards comme SGML ou plus récemment XML ont permis aux auteurs de documents classiques de capitaliser

leur travail lors de l'écriture d'une classe homogène de documents (par exemple des documents techniques) grâce à la notion de modèles de documents établis une fois pour toutes. Ils ont aussi amélioré les capacités de traitements automatiques des documents (conversion de formats cf. section 5.1, extraction d'information, formatage cf. section 5.2, etc.). De manière équivalente, la définition de modèles de documents multimédia aura un impact important sur la façon dont sont conçus puis traités ces documents. Des exemples de classes de documents multimédia sont aisément identifiables comme pour les guides touristiques, la présentation d'entreprises, la documentation technique multimédia, etc.).

Parmi les pistes explorées actuellement pour prendre en compte de façon efficace des modèles qui couvrent les aspects structuraux, spatiaux et temporels, les principes de formatage qui s'appuient sur des *schémas de transformation* (cf. section 4.2.1) paraissent prometteurs (cf. XSL). Cependant, un pas important devra être franchi pour que ces principes puissent être applicables dans le contexte de l'édition.

Ce thème constitue le cœur du sujet de thèse de Lionel Villard. Cette thèse est menée en collaboration avec la société Aérospatiale (laboratoire associé en ingénierie documentaire de la division AIRBUS) en vue d'expérimenter les concepts de classes de documents multimédia/hypermédia pour une documentation aéronautique d'exploitation. Une première étape de ce travail a permis de connecter l'environnement Madeus avec un fonds documentaire multimédia géré dans une base de données O2 [DRTV 1999].

6.4 Extension des modèles de représentation des documents

L'expérience acquise avec les langages comme XML, pour représenter les différents types de structures des documents selon un mode déclaratif, doit être mise à profit pour la description d'autres structures. De nombreux atouts sont à attendre de cette démarche : les possibilités de traitement obtenues grâce au typage de l'information, les possibilités d'intégration dans une même application de données de nature diverse grâce à un mode de description homogène, le partage d'outils de traitement de ces informations (parsers, outils de visualisation, etc.).

Deux études sur ce thème ont commencé dans le projet Opéra : l'une porte sur la spécification déclarative de relations spatio−temporelles les documents multimédia, l'autre sur les modèles et les traitements des informations audiovisuelles dans les systèmes multimédia. Suite à ces travaux, deux autres axes de réflexion émergent:

l'intégration de spécifications de type impératif dans les spécifications de type déclaratif et le besoin de spécification de documents adaptables.

A. Relations spatio−temporelles

Nous avons commencé par étendre le langage Madeus (cf. section 4.3.2) pour spécifier des relations spatio−temporelles comme par exemple, exprimer qu'un texte se met à défiler au moment où une vidéo est présentée. Ce type de spécification s'exprime le plus souvent sous forme de programmes (comme avec DHTML ou OpenScript de ToolBook). Nous avons préféré explorer l'approche déclarative qui répond mieux aux exigences de composition multimédia, d'édition et de portabilité. Ainsi un objet média peut être décomposé en sous−intervalles auxquels des attributs peuvent être attachés (durée, fonction de déplacement, etc.). Ces sous−intervalles peuvent être mis en relation avec d'autres objets ou sous−intervalles.

Ce premier travail doit être étendu et généralisé pour mieux combiner les spécifications de nature intrinsèquement déclarative comme « l'objet A est aligné à gauche avec l'objet B pendant la durée de l'objet C », de celles qui sont de nature événementielle comme « lorsque l'objet A atteint la position (x,y), alors démarrer l'objet B » (cf. point C ci−dessous).

B. Modèles et les traitements des informations audiovisuelles dans les systèmes multimédia

Les données audiovisuelles constituent une part de plus en plus importante de l'information véhiculée dans les applications multimédia à travers le web. Le plus souvent, la vidéo est considérée comme un objet de base (non décomposable) dans ces applications. Cependant, de nombreuses applications demandent de disposer d'une information de granularité plus fine et plus typée, par exemple pour l'indexation et la recherche par le contenu ou encore pour la composition avec d'autres objets média. Par ailleurs de nombreux travaux s'intéressent à l'analyse automatique des vidéos en vue d'en extraire le plus d'information sémantique possible.

Nous avons abordé ce sujet au travers du Dea de Tien Tran−Thuong dans lequel un premier modèle de description structurée des vidéos en XML a été proposé [Tran−Thuong 1999], [MRTV 1999]. Tien Tran−Thuong va poursuivre et étendre cette première expérience sur le thème des « modèles et traitements des informations audiovisuelles dans les systèmes multimédia ». C'est une activité de recherche qui se situe à la charnière des deux domaines pré−cités (applications multimédia et analyse de vidéos) et qui a pour objectif d'offrir non seulement des modèles de description pour la vidéo (en contribuant ainsi aux travaux des groupes de travail comme MPEG7

[MPEG 1999]) mais aussi des solutions algorithmiques pour tirer parti de ces modèles dans les applications (contrôle fin de la présentation, environnement de composition, filtre, etc.).

C. Modèle déclaratif versus modèle impératif

Comme j'ai tenté de le montrer tout au long de ce mémoire les approches de spécification déclarative présentent de nombreux atouts qui justifient leur emploi dans les systèmes de traitement de documents. Cependant, il semble que l'on arrive à une limite dans le pouvoir d'expression des modèles purement déclaratifs qui vient en partie du *comportement dynamique et réactif* des documents.

En effet, même avec les langages actuels (non génériques) qui utilisent une approche déclarative à base d'opérateurs temporels ou de contraintes temporelles, il est difficile de spécifier des documents ayant de véritables capacités d'interaction : action locale sur les objets (comme le changement de vitesse), réactivité contextuelle de la présentation. Par contre, ce type de comportement peut s'exprimer facilement sous forme d'expression conditionnelle ou sous forme de programmation par composition d'événements (langages de programmation, MHEG). Le problème est d'une part de faire coexister ce mode de spécification impérative avec une approche déclarative, et d'autre part de voir comment le prendre en compte au niveau générique.

D. Spécification de documents adaptables

Une autre extension à considérer vient des besoins de conception de documents qui soient adaptables (1) aux différents types de terminaux (du PC au téléphone mobile), (2) à différentes classes d'utilisateurs (du néophyte au spécialiste par exemple) et (3) au contexte réseau (débit et qualité de la transmission). Il est nécessaire d'étudier quelles informations doivent être fournies au moment de la spécification et comment les exploiter au moment de la présentation. Les traitements nécessaires à l'adaptation se situent à différentes étapes de la chaîne de production des documents : de façon statique, c'est−à−dire avant la présentation (choix d'une feuille de style par exemple), ou de façon dynamique, pendant la présentation. Ce problème rejoint donc la conception de systèmes de présentation et en particulier l'ordonnancement des présentations multimédia. Une approche que nous considérons également dans le projet est l'apport des techniques à base de contraintes pour ce problème (thèse de Frédéric Bès).

6.5 Édition de documents

L'activité de création de documents se doit d'être rendue la plus aisée possible grâce à l'outil informatique. Cet objectif *a priori* évident, et qui est la motivation première des expérimentations effectuées jusqu'à présent dans le projet Opéra, devient de plus en plus complexe à atteindre du fait de l'augmentation de la nature des informations à spécifier : typage, propriétés typographiques, propriétés temporelles, relations d'activation, paramètres pour l'adaptation, modèles de présentation, etc. Dans ce contexte, nous avons identifié un certain nombre d'axes de travail qui visent à améliorer les outils proposés aux auteurs :

1. Extension des techniques de formatage pour améliorer l'efficacité des algorithmes et permettre la prise en compte des informations spatiales et des informations temporelles.
2. Adaptation des approches de formatage par transformations successives au contexte particulier de l'édition (problème du maintien de la cohérence entre les informations source et les informations transformées).
3. Amélioration des algorithmes de vérification de cohérence des spécifications fournies par l'auteur.
4. Conception d'interfaces adaptées à la visualisation et à l'édition d'informations complexes et variées.
5. Proposition d'environnements graphiques pour la spécification de modèles de documents et de transformation de documents.

Ces différents axes sont abordés dans le cadre des thèses du projet Opéra qui sont en cours ou sur le point de débuter : Lionel Villard pour les points 1 et 2, Tien Tran−Thuong pour les points 1 et 4 pour le cas de la vidéo, Frédéric Bès pour les points 1 et 3, Laurent Tardif pour les points 3 et 4 et Emmanuel Pietriga pour le point 5, dans le cadre d'une thèse CIFRE effectuée avec le groupe DMTT (*Document Models and Transformation Technologies*) du Rank Xerox Research Centre.

Bibliographie

La liste de références bibliographiques ci−dessous est structurée en trois grandes parties : les références que je cite dans ce document, les standards et enfin mes publications.

[Publications]

[AF 1997] I. Atalli et P. Franchi−Zannettacci, ‘‘ A Document Manipulation System Based on Natural Semantics’’, *Mathematical and Computer Modelling*, 25(4), pp. 37−56, 1997.

[AFQ 1989] J. André, R. Furuta et V. Quint, *Structured documents*, Cambridge University Press, Cambridge, 1989.

[Akpotsui 1993] E. Akpotsui, *Transformation de types dans les sytèmes d’édition de documents structurés*, thèse spécialité Informatique, INPG, octobre 1993.

[Allen 1983] J. F. Allen, ‘‘Maintaining Knowledge about Temporal intervals’’, *CACM*, 26 (11), pp. 832−843, 1983.

[AM 1997] M. Adiba et F. Mocellin, ‘‘STORM: Une approche à objets pour les bases de données multimédias. STORM: An object−oriented approach for multimedia databases’’, *TSI − Technique et Science Informatiques − RAIRO* , 16(7), pp. 897−924, 1997.

[AN 1997] D. Adjeroh et K. Nwosu, ‘‘ Multimedia Database Management −− Requirements and Issues’’, *IEEE MultiMedia Magazine*, 4(3), pp. 24−33, 1997.

[AQ 1992] E. Akpotsui et V. Quint, ‘‘Type Transformation in Structured Editing Systems’’, *Proc. of the EP’92 Conference*, édité par G. Coray & C. Vanoirbeek, pp. 27−41, C. Cambridge University Press, avril 1992.

[AR 1997] J. André et H. Richy, ‘‘Hypertextes ou documents structurés ? − Étude de cas en critique génétique’’, *Hypertextes et Hypermédias*, édité par Hermès, pp. 13−26, Paris, septembre 1997.

[Arnon 1993] D.S. Arnon, ‘‘Scrimshaw: A Language for Document Queries and Transformations’’, *Electronic Publishing − Origination, Dissemination and Design, Proc. of the EP’94 Conference*, vol. 6, num. 4, pp. 385−396, décembre 1993.

[Balise 1996] *Balise 3 Reference Manual*, AIS S.A., 1996.

[Bétrancourt 1996] M. Bétrancourt, *Facteurs spatiaux et temporels dans le traitement cognitif des complexes texte−figure*, Doctorat de sciences cognitives, INPG, octobre 1996.

[BF 1995] A. Borning, B. Freeman−Benson, ‘‘The OTI Constraint Solver: A Constraint Library for Constructing Interactive Graphical User’’, *In Proc. of the First International Conference on Principles and Practice of Constraint Programming*, Cassis, France, pp. 624−628, septembre 1995.

[Blasselle 1997] B. Blasselle, *Histoire du livre − Volumes I et II*, Découvertes Gallimard, 1998.

[BLM 1997] A. Borning, R. Lin et K. Marriott, ‘‘Constraints for the Web’’, *In Proceedings of the ACM Multimedia Conference*, novembre 1997.

[Bonhomme 1998] S. Bonhomme, *Transformations de documents structurés : une combinaison des approches déclaratives et automatiques*, Doctorat d’informatique, Université Joseph Fourier, décembre 1998.

[BTP 1998] M. Bétrancourt, L. Tardif et A. Pellegrin, ‘‘Conception et évaluation d’un outil d’édition et présentation de documents multimédia’’, *Colloque ERGO’IA 98*,, pp. 69−78, Biarritz, novembre 1998.

[BZ 1992] C. Buchanam et P. T. Zellweger, ‘‘Specifying Temporal Behavior in Hypermedia Documents ’’, *Proc. of the ACM Conference on Hypertext*, pp. 262−271, décembre 1992.

[Carcone 1997] L. Carcone, *Formatage spatial dans un environnement d’édition/présentation de documents multimédia*, Mémoire d’ingénieur Cnam, décembre 1997.

[CCS 1998] V. Christophides, S. Cluet et J. Siméon, ‘‘Semistructured and Structured Integration Reconciled: YAT += Efficient Query Processing’’, *publication interne*, http://www−rocq.inria.fr/~simeon/YAT/, novembre 1998.

[CGHIP 1994] S. Chawathe, H. Garcia−Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman et J. Widom, ‘‘The TSIMMIS Project: Integration of Heterogeneous Information Sources.’’, *Proc. of IPSJ Conference*, pp. 7−18, Tokyo, octobre 1994.

[Claves 1995] Ph. Claves, *Restructuration dynamique de documents structurés*, Mémoire d’ingénieur Cnam, mars 1995.

[DK 1995] A. Duda et C. Keramane, ‘‘Structured Temporal Composition of Multimedia Data’’, *Proc. of IEEE International Workshop on Multi−Media Data Base Management Systems*, Blue Mountain Lake, NY, août 1995.

[DR 1996] D. Decouchant et M. Romero Salcedo, ‘‘Alliance: a Structured Cooperative Editor on the Web’’, *CSCW and the Web, Proc. of the 5th ERCIM/W4G Workshop*, édité par U. Busbach, D. Kerr, K. Sikkel, pp. 7−12, Arbeitspapiere der GMD 984, Sankt Augustin, février 1996.

[Digithome 1996] *Introduction to IDM (Intelligent Document Manager)*, Digithome Electronic Publishing, Dublin, Ireland, 1996, http://www.digithome.com.

[English 1998] J. English, *Cost 2 Reference Manual*, 1998.

[Freeman−Benson 1993] B. Freeman−Benson, ‘‘Converting an Existing User Interface to Use Constraints’’, *Proceeding of the ACM SIGGRAPH Symposium on User Interface Software and Technology*, pp. 207−215, Atlanta, Georgia, novembre 1993.

[François 1997] P. François, *Etude et conception d’une structure d’accueil de documents hypermédia structurés selon les normes SGML et HyTime. Application à la documentation aéronautique*, Thèse de doctorat, Université Paul Sabatier, Toulouse, octobre 1997.

[Fron 1994] A. Fron, *"Programmation par contraintes"*, Ed. Addison−Wesley, Paris, avril 1994.

[FQA 1988] R. Furuta, V. Quint et J. André, ‘‘Interactively Editing Structured Documents’’, *Electronic Publishing*, 1(1), pp. 19−44, avril 1988.

[FS 1988] R. Furuta et P.D. Stotts, ‘‘Specifying Structured Document Transformations’’, *Proc. of the EP’88 Conference*, édité par Cambridge University Press, Nice (France), 20−22 avril 1988.

[GQV 98] R. Guetari, V. Quint et I. Vatton, ‘‘Amaya: an Authoring Tool for the Web’’, *MCSEAI’98 International Conference*, Tunisie, décembre 1998.

[Hardman 1998] L. Hardman, *Modelling and Authoring Hypermedia Documents*, Doctorat d’université, Université d’Amsterdam, Pays−Bas, mars 1998.

[Inria Rhône−Alpes 1998] *Projet Storia : Réseau français d'expérimentation industrielle de solutions pour les autoroutes de l'information*, 1998, http://www.inrialpes.fr/storia.html.

[Inria 1998] *Rapport d'activité du projet Opéra*, Inria, Domaine de Voluceau, Rocquencourt, 1998, http://www.inria.fr.

[KS 95] M. Y. Kim et J. Song, ''Multimedia Documents with Elastic Time'', *Proc. of ACM Multimedia'95 Conference*, pp. 143−154, San Francisco, novembre 1995.

[LMR 1998] J. Le Maitre, E. Murisasco et M. Rolbert, ''From annotated Corpora to Databases : the SgmlQL language'', *CSLI lectures notes*, n. 77, janvier 1998.

[LS 1996a] N. Layaïda et L. Sabry−Ismail, ''Maintaining Temporal Consistency of Multimedia Documents using Constraint Networks'', *Multimedia Computing and Networking 1996*, édité par M. Freeman, P. Jardetzky, H. M. Vin, pp. 124−135, SPIE 2667, janvier 1996.

[LS 1996b] N. Layaïda et L. Sabry−Ismail, ''MADEUS : un modèle de document multimédia structuré'', *Techniques et Sciences de l'Informatique (TSI)*, 15(9) (numéro thématique "Multimédia Collecticiels"), 1996.

[Layaïda 1997] N. Layaïda, *Madeus: un système d'édition et de présentation de documents structurés multimédia*, Thèse de doctorat, Université Joseph Fourier, juin 1997.

[Macromedia 1998] Macromedia, *Flash and Director*, 1998.

[MAGQW 1997] J. McHugh, S. Abiteboul, R. Goldman, D. Quass et J. Widom, ''Lore: A Database Management System for Semistructured Data'', *SIGMOD Record*, 26(3), pp. 54−66, 1997.

[Marchiori 1998] M. Marchiori, ''QL'98 − The Query Languages Workshop'', W3C, Boston, décembre 1998.
http://www.w3.org/TandS/QL/QL98/program.html.

[Meyer−Boudnik 1995] T. Meyer−Boudnik et W. Eeffelsberg, ''MHEG Explained'', *IEEE MultiMedia Magazine*, 2(1), pp. 26−38, 1995.

[MKNS 1989] S.A. Mamrak, M.J. Kaelbling, C.K. Nicholas et M. Share, ''Chameleon: A System for Solving the Data−Translation Problem'', *IEEE Transactions on Software Engineering*, pp. 1090−1108, septembre 1995.

[Montarnal 1997] C. Montarnal, *L'activité de linéarisation lors de la production de textes descriptifs*, Doctorat de sciences cognitives , INPG, décembre 1997.

[Myers 1990] B.A. Myers, ''Taxonomies of visual programming and program visualization'', *Journal of Visual Languages and Computing*, 1 pp. 97−123, mars 1990.

[Nanard 1995] J. Nanard et M. Nanard, ''Hypertext design environments and the hypertext design process'', *Communications of the ACM*, 38(8), pp. 49−56, août 1995.

[Opéra 1997] Projet Opéra, *THOT, A structured document editor*, Inria, 1997, http://www.inrialpes.fr/opera/Thot/.

[Quint 1987] V. Quint, *Une approche de l'édition structurée des documents*, thèse d'état, Université scientifique, technologique et médicale de Grenoble, mai 1987.

[Quint 1994] V. Quint, translated by E. Munson, *The languages of Thot*, INRIA, 655 av. de l'Europe, 38330 Montbonnot − France, 1994.

[QV 1994] V. Quint et I. Vatton, ''Active Documents as a Paradigm for Human−Computer Interaction'', *Workshop on Software Engineering and Human−Computer Interaction: Joint Research Issues, Preprints*, édité par R. Taylor and J. Coutaz, pp. 255−262, Sorrento, Italy, mai 1994.

[QV 1997] V. Quint et I. Vatton, ''An Introduction to Amaya.'', *World Wide Web Journal*, 2(2), pp. 39−46, printemps 1997.

[RJMB 1993] G. van Rossum, J. Jansen, K. Mullender et D. Bulterman, ''CMIFed : a presentation Environment for Portable Hypermedia Documents'', *Proc. of the ACM Multimedia Conference*, California, 1993.

[Romero 1998] M. Romero Salcedo, *Alliance sur Internet : support pour l'édition coopérative de documents structurés sur un réseau à grande distance*, Doctorat Informatique, Institut National Polytechnique de Grenoble, 29 avril 1998.

[SMFB 1993] M. Sannella, J. Maloney, B. Freeman−Benson, and A. Borning, ''Multi−way versus One−way Constraints in User Interfaces: Experience with the DeltaBlue Algorithm'', *Software−Practice and Experience*, 32(5), pp. 529−566, mai 1993.

[Sabry 1999] L. Sabry−Ismail, *Schéma d'exécution pour les documents multimédia distribués*, Doctorat d'informatique, Université Joseph Fourier, janvier 1999.

[SDLS 1996] P. Sénac, M. Diaz, A. Léger et P. De Saqui−Sannes, ''Modeling Logical and Temporal Synchronization in Hypermedia Systems'', *IEEE Journal of Selected Areas on Communications*, 14(1), pp. 84−103, 1996.

[Stemler 1997] L. K. Stemler, ''Educational Characteristics of Multimedia: a Literature Review'', *Journal of Educational Multimedia and Hypermedia*, 6(3/4), pp. 229−359, 1997.

[Stern 1997] Y. Stern, ''Les quatre dimensions des documents électroniques'', *Document numérique*, 1(1), pp. 55−60, 1997.

[Tardif 1997] L. Tardif, *Visualisation du scénario temporel d'un document multimédia*, DEA d'nformatique de l'Université Joseph Fourier, Grenoble, juin 1997.

[TEI 1998] TEI, *Text Encoding Initiative*, University of Illinois at Chicago, 1940 W. Taylor St., Room 124 Chicago, IL 60612−7352, USA, 1998. http://www.uic.edu/orgs/tei/index.html.

[Tran−Thuong 1999] T. Tran Thuong, *Description de la structure des vidéos pour les applications multimédias*, Rapport de DEA Image Vision et Robotique de l'Université Joseph Fourier (Grenoble 1) et de l'INPG, juin 1999.

[Trombettoni 1997] G. Trombettoni, *Algorithmes de maintien de solution par propagation locale pour les systèmes de contraintes*, Doctorat d'informatique, Université de Nice Sophia−Antipolis, juin 1997.

[Villard 1998] L. Villard, *Spécification de relations spatio−temporelles dans un document multimédia*, DEA d'nformatique de l'Université Joseph Fourier, Grenoble, juin 1998.

[Weissman 1995] L. Weissman, *The Architecture of Information*, Thèse de doctorat, Massachussets Institute of Technology, février 1995.

[Standards et Recommandations]

[CCETT 1998] CCETT, *MhegDitor*, http://www.ccett.fr/mheg/converter.htm, 1998.

[ISO 1986] ISO 8879, *Standard Generalized Markup Language (SGML)*, ISO, http://www.oasis−open.org/cover/sgml−xml.html, 1986.

[ISO 1996] ISO, *ISO/IEC 10179:1996 Document Style Semantics and Specification Language (DSSSL)*, ftp://ftp.ornl.gov/pub/sgml/WG8/DSSSL/, ISO/IEC 1996.

[ISO 1997] ISO/IEC JTC1/SC18/WG8 N1920, *Hypermedia/ Time−based Structuring Language (HyTime), Second edition*, ISO/IEC, août 1997. http://www.ornl.gov/sgml/wg8/docs/n1920/html/n1920.html.

[MPEG 1999] *MPEG−7 Document*, http://www.darmstadt.gmd.de/mobile/MPEG7/Documents.htm, 1999.

[W3C Namespaces 1999] W3C Recommendation, *Namespaces in XML*, http://www.w3.org/TR/1998/REC−xml−names, 14−January 1999.

[W3C SVG 1998] W3C Working Draft, *Scalable Vector Graphics Language (SVG) Version 1.0*, http://www.w3.org/TR/WD−svg, 1998.

[W3C XSL 1999] W3C Working Draft, *Extensible Stylesheet Language (XSL) Specification*, http://www.w3.org/TR/WD−xsl, 21−April 1999.

[W3C XML 1998] W3C Recommendation, *Extensible Markup Language (XML) 1.0*, http://www.w3.org/TR/1998/REC−xml−19980210, 10−February 1998.

[W3C XML−QL 1998] W3C Submission, *XML−QL: A Query Language for XML*, http://www.w3.org/TR/1998/NOTE−xml−ql−19980819/, 19−August 1998.

[W3C XLink 1999] W3C Working Draft, *XML Linking Language (XLink)*, http://www.w3.org/TR/xlink, 26 July 1999.

[W3C XPointer 1998] W3C Working Draft, *XML Pointer Language (XPointer)*, http://www.w3.org/TR/WD−xptr, 9 July 1999.

[W3C XML Schema 1999] W3C Note, *XML Schema Requirements*, http://www.w3.org/TR/NOTE−xml−schema−req, 15−February 1999.

[W3C MathML 1998] W3C Recommendation, *Mathematical Markup Language (MathML) 1.0 Specification*, http://www.w3.org/TR/REC−MathML/, 07−April 1998.

[W3C HTML 1998] W3C Recommendation, *HTML 4.0 Specification*, http://www.w3.org/TR/REC−html40/, 24−April 1998.

[W3C RDF 1998] W3C Recommendation, *RDF Specification*, http://www.w3.org/TR/REC/, 1998.

[W3C CSS 1998] W3C Recommendation, *Cascading Style Sheets, level 2, CSS2 Specification*, http://www.w3.org/TR/REC−CSS2/, 12−May 1998.

[W3C SMIL 1998] W3C Recommendation, *Synchronized Multimedia Integration Language (SMIL) 1.0 Specification*, http://www.w3.org/TR/REC−smil, 15−June 1998.

[W3C WebDAV 1998] W3C Working Draft, *(WebDAV)*, http://www.w3.org/TR/WD−WebDAV, 1998.

[**Bibliographie personnelle classée par année**]

[Roisin 1985] C Roisin, ‘‘Protocol Description with the OCCAM Language’’, *5th Workshop on Protocol Specification, Verification and Testing*, North−Holland, juin 1985.

[HR 1986] S. Haj Houssain et C. Roisin, ‘‘OSI Server for Local Area Networks’’, *Symposium International sur les Systèmes de Communication Locaux et PBX*, North−Holland, novembre 1986.

[Balter 1991] R. Balter, J. Bernadat, D. Decouchant, A. Duda, A. Freyssinet, S. Krakowiak, M. Meysembourg, P. Le Dot, H. Nguyen Van, E. Paire, M. Riveill, C. Roisin, X. Rousset de Pina, R. Scioville, G. Vandôme, ‘‘Architecture and Implementation of Guide, an Object−Oriented Distributed System’’, *Computing Systems*, 1991.

[DDRRR 1991] D. Decouchant, P. Le Dot, M. Riveill, C. Roisin, X. Rousset de Pina, ‘‘A Synchronization Mechanism for an Object Oriented Distributed System’’, *11th International Conference on Distributed Computing Systems*, Arlington, Texas, 20−24 mai 1991.

[JRS 1991] H. Jamrozik, C. Roisin, M. Santana, ‘‘A Graphical Debugger for Object−Oriented Distributed Programs’’, *TOOLS’USA 91*, Santa Barbara, California, juillet 1991.

[RS 1991] C. Roisin, M. Santana, *The Observer: A Tool for Observing Object−Oriented Distributed Applications*, (8−91), Rapport Bull−IMAG, janvier 1991.

[AQR 1992] E. Akpotsui, V. Quint et C. Roisin, ‘‘Type Modelling for Document Transformation in Structured Editing Systems’’, *Workshop on Principles of Document Processing*, Washington, octobre 1992.

[RV 1993] C. Roisin et I. Vatton, *Formatting Structured Documents*, (R.R. 2044), INRIA, septembre 1993.

[AR 1994] E. Akpotsui et C. Roisin, ‘‘Implementing the Cut−and−Paste Operation in a Structured Editing System’’, *Workshop on Principles of Documents Processing*, édité par M. Murata and H. Gallaire, Darmstadt, avril 11−12 1994.

[RV 1994] C. Roisin et I. Vatton, ‘‘Merging Logical and Physical Structures in Documents’’, *Electronic Publishing −− Origination, Dissemination and Design, special issue: Proc. of the EP’94 Conference*, 6(4), pp. 327−337, avril 1994.

[QRV 1995] V. Quint, C. Roisin et I. Vatton, ‘‘A Structured Authoring Environment for the World−Wide Web’’, *Computer Networks and ISDN Systems*, 27(6), pp. 831−840, avril 1995.

[BR 1996] S. Bonhomme et C. Roisin, ‘‘Interactively Restructuring HTML Documents’’, *Computer Network and ISDN Systems*, 28(7−11), pp. 1075−1084, mai 1996.

[AQR 1997] E. Akpotsui, V. Quint et C. Roisin, ‘‘Type Modelling for Document Transformation in Structured Editing Systems’’, *Mathematical and Computer Modelling*, 25(4), pp. 1−19, 1997.

[BQRRV 1997] S. Bonhomme, V. Quint, H. Richy, C. Roisin et I. Vatton, *The Thot User’s Manual*, Inria Rhône−Alpes, http://www.inrialpes.fr/opera/Thot/Doc/Thotman−E.html, mai 1997.

[BR 1997] S. Bonhomme et C. Roisin, ‘‘Transformations de documents électroniques’’, *Document Numérique*, 1(4), décembre 1997.

[CJR 1997] L. Carcone, M. Jourdan et C. Roisin, ‘‘Présentation de documents multimédia basée sur les contraintes’’, *Workshop on Electronic Page Models, LAMPE*, 22−23 septembre 1997.

[JLRS 1997] M. Jourdan, N. Layaïda, C. Roisin et L. Sabry−Ismail, ‘‘An Integrated Authoring and Presentation Environment for Interactive Multimedia Documents’’, *Proc. of the 4th Conference on Multimedia Modelling*, World Scientific Publishing, Singapore, 18−19 novembre 1997.

[JRT 1997] M. Jourdan, C. Roisin et L. Tardif, ‘‘User Interface of a new generation of authoring environment of multimedia documents’’, *Proc. of the Third ERCIM Workshop on User Interfaces for All*, Strasbourg (France) , novembre 1997.

[RCA 1997] C. Roisin, P. Claves et E. Akpotsui, ‘‘Implementing the Cut−and−Paste Operation in a Structured Editing System’’, *Mathematical and Computer Modelling*, vol. 26(1), pp. 85−96, 1997.

[SR 1997] M. Serrano et C. Roisin, *Storia : bilan et rapport final de la phase 1 de l'expérimentation. Volet 1 : édition coopérative et publication de documents sur le Web*, INRIA Rhône−Alpes, septembre 1997.

[SRL 1997] L. Sabry−Ismail, C. Roisin et N. Layaïda, ''Navigation in Structured Multimedia Documents using Presentation Context'', *Hypertextes et Hypermédias*, édité par Hermès, Paris, septembre 1997.

[JLR 1998] M. Jourdan, N. Layaïda et C. Roisin, *A survey on authoring techniques for temporal scenarios of multimedia documents*, Handbook of Internet and Multimedia Systems and Applications, part 1: Tools and Standards, CRC Press, avril 1998.

[JRT 1998a] M. Jourdan, C. Roisin et L. Tardif, ''Édition et Visualisation Interactive de Documents Multimedia'', *Proc. of Electronic Publishing'98, St Malo, LNCS n.1375*, pp. 370−380, Springer, avril 1998.

[JRT 1998b] M. Jourdan, C. Roisin et L. Tardif, ''Constraints Techniques for Authoring Multimedia Documents'', *ECAI 98 Workshop on Constraints for artistic applications*, Brighton (UK), août 1998.

[JRT 1998c] M. Jourdan, C. Roisin et L. Tardif, ''Multiviews Interfaces for Multimedia Authoring Environments'', *Proceedings of the 5th Conference on Multimedia Modeling*, pp. 72−79, IEEE Computer Society, Lausanne, octobre 1998.

[JLRST 1998] M. Jourdan, N. Layaïda, C. Roisin, L. Sabry−Ismail et L. Tardif, ''Madeus, an Authoring Environment for Interactive Multimedia Documents'', *Proc. of the ACM Multimedia'98 Conference*, pp. 267−272, Bristol (UK), septembre 1998.

[Roisin 1998] C. Roisin, ''Authoring Structured Multimedia documents'', *25th Conference on Current Trends in Theory and Practice of Informatics (SOFSEM'98)*, édité par B. Rovan, pp. 222 −239, Springer, LNCS 1521, Jasna, Slovakia, novembre 1998.

[JRT 1998d] M. Jourdan, C. Roisin et L. Tardif, *Rapport de conception de l'architecture du logiciel Madeus version 2.0*, Rapport de contrat, INRIA Rhône−Alpes, juin 1998.

[CR 1998] L. Carcone et C. Roisin, *Storia : bilan et rapport final de la phase 2 de l'expérimentation. Volet 1 : édition coopérative et publication de documents sur le Web*, INRIA Rhône−Alpes, décembre 1998.

[BR 1999] S. Bonhomme et C. Roisin, ''Transformation de structures XML'', *Le Micro Bulletin Thématique : L'information scientifique et technique et l'outil Internet − Cnrs*, (3), pp. 145−160, mai 1999.

[JLR 1999] M. Jourdan, N. Layaïda et C. Roisin, ''le temps dans les documents'', *Techniques de l'ingénieur*, 249, rue de Crimée, Paris, janvier 1999.

[JRT 1999a] M. Jourdan, C. Roisin et L. Tardif, ''Constraints Techniques for Authoring Multimedia Documents'', *Constraints Journal*, à paraître 1999.

[JRT 1999b] M. Jourdan, C. Roisin et L. Tardif, ''A Scalable Toolkit for Designing Multimedia Authoring Environments'', *numéro spécial 'Multimedia Authoring and Presentation: Strategies, Tools, and Experiences' de Multimedia Tools and Applications Journal*, à paraître, 1999.

[JRTV 1999] M. Jourdan, C. Roisin, L. Tardif et L. Villard, ''Authoring SMIL documents by direct manipulations during presentation'', *World Wide Web, Balzer Science Publishers*, à paraître, 1999.

[LRS99] N. Layaïda, C. Roisin et L. Sabry−Ismaïl, *Support d'exécution de documents multimédia*, Chapitre 8, Hermès Science Publications, Paris, à paraître, 1999.

[RTV 1999] C. Roisin, T. Tran Thuong et L. Villard, ''Integration of structured video in a multimedia authoring system'', *Workshop EG Multimedia'99*, Milan, septembre 1999.

[DRTV 1999] F. Duluc, C. Roisin, L. Tardif et L. Villard, ''Un système multimédia complet pour la documentation technique aéronautique'', soumis, mai 1999.

[SLR 1999] L. Sabry−Ismail, N. Layaïda et C. Roisin, ''Dealing with incertain durations in synchronized multimedia presentations'', publication interne, 1999.

[CJRTV 1999] Y. Carbonneaux, M. Jourdan, C. Roisin, L. Tardif et L. Villard, *Architecture détaillée du logiciel Madeus2.1*, Rapport de contrat, INRIA Rhône−Alpes, janvier 1999.

[MRTV 1999] R. Mohr, C. Roisin, T. Tran Thuong et L. Villard, *Rapport de description XML des structures de vidéo*, Rapport de contrat, INRIA Rhône−Alpes, mars 1999.

Résumé

Ce mémoire décrit les activités de recherche que j'ai menées depuis ma thèse et plus particulièrement celles que j'effectue dans le projet Opéra (outils pour les documents électroniques : recherche et applications) de l'INRIA sur les documents structurés multimédia. Ce document tente d'inscrire les différents travaux réalisés dans un cadre plus large qui comprend d'une part la modélisation des documents et d'autre part les applications de traitement de documents. Les principales contributions sont les suivantes :

- Représentation abstraite des types de documents de façon à permettre de transformer les structures de documents.
- Représentation de la structure logique et de la structure spatiale, ou comment intégrer des structures spatiales et logiques de façon à formater les documents structurés selon des modèles de pages complexes.
- Représentation de l'information temporelle des documents multimédia qui s'intègre aux représentations logiques et spatiales et qui soit adaptée à la construction d'environnements d'édition.

Mots−clés

Document structuré, multimédia, transformation de structure, formatage, édition.

Abstract

This document describes my research activities carried out in the project Opéra at INRIA on multimedia and structured documents. These activities are presented in a more general context including both document modeling and authoring applications. The main contributions are the following:

- An abstract representation of document type that meets document transformation needs in an authoring environment.
- A complex formatting method for structured documents that merges logical structures and physical structures.
- A representation of the temporal organization of multimedia documents that integrates well into the structured document model and allows the creation of powerful authoring environments.

Keywords

Structured document, multimedia, structure transformation, formatting, authoring.

Habilitation à diriger les recherches

spécialité
INFORMATIQUE

Présentée à
L'INSTITUT NATIONAL POLYTECHNIQUE
DE GRENOBLE

par
Cécile Roisin

Documents structurés multimédia

1 Des réseaux aux documents 5
1.1 Réseaux d'ordinateur et la spécification des protocoles de communication (1985−1991) 6
1.2 Langages à objets pour les systèmes répartis (1985−1991) 6
1.3 Documents structurés multimédia (depuis 1991) 7
2 Plan de ce mémoire 9
3 Qu'est−ce qu'un document ? 11
3.1 Quelles définitions ? 11
3.2 Quels sont les problèmes liés aux traitements électroniques des documents ? 14
4 Modèles de documents 19
4.1 Modèles de représentation des structures logiques 20
4.1.1 Principes et langages 20
4.1.2 Modèle de type de documents pour les transformations de structures 22

4.2 Modèles de représentation des structures spatiales 25
4.2.1 Principes et langages 26
4.2.2 Le langage P et les modèles de pages pour les documents structurés 28
4.2.3 Positionnement spatial de documents multimédia basé sur les contraintes 32
4.3 Modèles de représentation des structures temporelles 34
4.3.1 Principes et caractéristiques des documents multimédia 34
4.3.1.1 Nature des objets multimédia 35
4.3.1.2 Style et activation 35
4.3.1.3 Composition temporelle 35
4.3.2 Le modèle de documents de Madeus 37
4.4 Modèles de représentation des structures de navigation dans les documents 40
4.4.1 Introduction 40
4.4.2 Principes de navigation temporelle 41

5 Applications documentaires 45

5.1 Transformation des documents structurés 45
5.1.1 Étude des besoins 46
5.1.2 Techniques de transformation 47
5.1.3 Bilan 50
5.2 Outils auteur de documents structurés 50
5.2.1 Thot 50
5.2.2 Ma participation à Thot 51
5.2.3 Travaux construits à partir de Thot 51
5.2.4 Bilan 52
5.3 Environnements auteur de documents multimédia 53
5.3.1 Les problèmes posés par l'édition de documents multimédia . . 53
5.3.2 Environnement auteur de documents multimédia 54
5.3.3 Présentation de documents multimédia 56
5.3.4 Bilan 58

6 Orientations de recherche dans le domaine des documents 59

6.1 Perspectives générales 59
6.2 Axes de travail du projet Opéra 59
6.3 Modèles génériques de documents multimédia 60
6.4 Extension des modèles de représentation des documents 61
6.5 Édition de documents 64

Bibliographie