



Conclusion

1. Rappel des objectifs et contexte

Le projet Opéra étudie les problèmes posés par la conception, l'édition et la présentation des documents multimédias. Mon travail de thèse effectué dans cette équipe, a concerné la conception et la réalisation d'un système d'édition qui permet de prendre en compte, pendant le processus d'édition, les besoins des auteurs, novices ou experts.

La construction d'un environnement d'édition est soumise à des contraintes multiples. Ces contraintes sont liées d'une part à la multitude de formats de spécification de documents multimédias, et d'autre part aux différents objectifs et aux compétences des auteurs.

La plupart des systèmes d'édition actuels proposent des méthodes d'édition très liées au format de document sous-jacent. De ce fait, l'édition ne propose pas, ou peu, de fonctionnalités pour aider l'auteur et automatiser certaines tâches.

2. Démarche de travail et bilan

Mon travail de thèse s'est organisé en deux grandes parties, ce découpage se retrouve dans la thèse :

Environnements auteur de documents multimédias

La première partie est axée autour des environnements auteur de documents multimédias. J'ai tout d'abord analysé la situation courante des différents environnements auteur. Au cours de cette analyse, j'ai dégagé quatre principaux formalismes d'édition de documents multimédias, et j'ai montré comment les environnements auteur permettaient de les éditer.

Le premier résultat de la thèse consiste, à partir de cette étude, à la définition de ce que serait un environnement d'édition permettant à un auteur de spécifier à la fois simplement et de manière complète son document multimédia.

Le deuxième résultat présenté est la boîte à outils Kaomi. Elle doit permettre de construire de tels environnements auteur, tout en offrant aux concepteurs de ces environnements une mise en commun des services.

Systèmes de contraintes dans l'édition de documents multimédias :

A partir de ces résultats et des différentes expériences menées, je me suis plus particulièrement intéressé plus particulièrement aux contraintes et aux systèmes de résolution de contraintes. L'étude des résolveurs de contraintes constitue la deuxième partie de la thèse. Dans une première étape, j'ai dégagé un ensemble de besoins qui n'étaient à ce jour pas satisfaits. J'ai ensuite démontré que les technologies basées sur les contraintes pouvaient apporter des réponses intéressantes aux besoins liés à l'édition de documents multimédias. Pour cela, j'ai étudié les différents mécanismes de résolution

ainsi que leur utilisation optimale dans le contexte de l'édition de documents multimédias.

Ainsi, le troisième résultat de la thèse consiste en l'évaluation des différents types de solveurs dans notre contexte. Différentes expériences m'ont permis d'obtenir des résultats qualitatifs et quantitatifs sur les différentes approches de résolution de contraintes dans notre contexte d'édition de documents multimédias. À partir de cette analyse, j'ai proposé un algorithme hybride offrant un bon compromis entre les performances et la qualité des différentes réponses calculées.

Le quatrième résultat de la thèse est l'implémentation d'environnements auteur basés sur la boîte à outils Kaomi. Ces environnements auteur sont représentatifs des différents formalismes d'édition et de spécification de documents multimédias mais aussi de contextes applicatifs différents comme l'édition de workflow ou la gestion de bases documentaires. Ces expériences ont permis de valider d'une part la genericité de la boîte à outils, et d'autre part l'apport des contraintes au travers des services d'édition avancés qui ont été mis en oeuvre.

Enfin, à partir de ces expériences, j'ai dégagé les différentes utilisations futures des techniques à base de contraintes pour le contexte de l'édition de documents multimédias.

3. Perspectives

J'ai choisi de classer les perspectives de ce travail selon trois axes. Le premier axe concerne les apports possibles de l'utilisation des contraintes dans un contexte d'édition de documents multimédias de manière à exploiter pleinement le travail réalisé dans cette thèse. Dans un deuxième temps, je présenterai la situation de la boîte à outils Kaomi vis-à-vis des évolutions des documents multimédias. Et enfin, dans un troisième temps, je présenterai les différents points liés aux contraintes, qui doivent être améliorés pour rendre cette technologie complètement exploitable dans un contexte applicatif.

3.1 Perspectives sur l'utilisation des technologies contraintes

Dans le cadre de l'édition de documents multimédias, les perspectives d'utilisation des technologies contraintes sont nombreuses. Dans un premier temps, je vais présenter les différentes utilisations possibles de ces techniques qui sont des applications directes des travaux réalisés, je présenterai ensuite une problématique plus générale sur l'adaptation de documents qui se développe aujourd'hui du fait de l'essor de la téléphonie mobile et des ordinateurs de poche.

- **Prototypage rapide de documents**

Un des besoins essentiels lors de la conception de documents est la possibilité de prototyper rapidement une version de manière à pouvoir avoir un aperçu du document [Bollard99]. Dans ce contexte, un formalisme de haut niveau à base de contraintes permet de spécifier de manière partielle un document. L'auteur, en peu de temps, peut obtenir une maquette qui lui permet d'avoir une idée précise de ce que sera son document dans sa forme finale et même de naviguer dans un espace de solutions. Cette phase de prototypage est très utile lors de la production de documents pour, par exemple, avoir la possibilité de proposer des maquettes différentes aux autres auteurs, aux collaborateurs ou même aux clients. Lesquelles serviront ensuite de point de départ pour les discussions entre les concepteurs.

- **Effets de style ou macros adaptables**

Dans le cadre des documents multimédias, il arrive que l'on ait besoin de définir un effet de présentation pour un ensemble de n objets. Que se passe-t-il, lorsque l'on constate qu'il faut insérer un objet de plus dans cet effet ? Par exemple, l'auteur définit une animation de cinq objets qui s'animent

pour se regrouper et former un titre. Si l'auteur désire insérer un sixième objet dans son animation, il est souhaitable qu'il n'ait pas à recalculer le placement et le mouvement de tous ces objets. Les techniques à base de contraintes peuvent être utiles pour calculer une nouvelle solution. En effet, cela permet au programmeur de l'effet de style de ne pas écrire toutes les solutions possibles dont un utilisateur aura besoin. Il décrira seulement les contraintes à respecter entre les objets et laissera le calcul du placement exact des objets au résolveur de contraintes.

● **Adaptation de document et contraintes**

Un des intérêts d'utiliser des techniques à base de contraintes est de permettre le formatage, soit statiquement au moment de l'édition du document, soit au moment de la présentation. Ce dernier cas permet d'adapter le document au lecteur ou aux conditions de présentation.

Je présente ici les principales utilisations de ces possibilités de formatage.

● **Adaptation de documents générés**

Un des intérêts de l'utilisation de schémas prédéfinis (chapitre II section 4.1) est la simplicité de spécification pour l'auteur. Un des intérêts d'une technologie contraintes combinée avec les schémas prédéfinis est de rendre plus flexible ces schémas. Imaginons par exemple un schéma pour présenter en séquence une liste de photographies. Cette liste peut être le résultat d'une requête vers une BD, ou une liste de photographies fournies par l'auteur. L'utilisation de résolveurs et d'un formalisme de spécification à base de relations permet au lecteur de fixer la durée globale de la présentation et le système déduira automatiquement la durée de présentation pour chacune des images de la séquence en fonction de leur nombre.

● **Multi-langages**

Une des applications intéressantes qui tire parti d'un formatage souple est de pouvoir réaliser des documents multilingues, formatés en fonction de la langue. Par exemple, un média audio n'aura pas la même durée en anglais ou en français. De la même façon, un texte anglais n'aura pas la même longueur que sa traduction française. De ce fait, avoir un document contenant de la flexibilité et un outil permettant de formater ce document est un apport intéressant pour l'auteur, qui laisse ainsi le soin au formateur de calculer au mieux le placement des objets du document, en concervant si possible la durée intrinsèque de ces médias. L'auteur ne spécifie ainsi qu'un document. Le système le formatera alors de façon optimale pour chacune des langues.

● **Multi-lecteur**

On peut, de la même manière, adapter un document en fonction des capacités du lecteur. Par exemple, dans le cadre de l'enseignement, on peut concevoir des présentations différentes d'un même cours en fonction de l'élève et/ou de son niveau d'apprentissage. De la même manière, dans le domaine médical, le dossier d'un patient doit être présenté de manière différente à un spécialiste, à l'infirmière traitante ou au patient lui-même. La visualisation ou non de certains objets va donc entraîner une phase d'adaptation du document qui pourra être prise en charge par le résolveur de contraintes.

● **Formatage dynamique**

Une autre forme d'adaptation qui peut être traitée par les contraintes est l'adaptation dynamique du document au cours de la présentation. Cette adaptation se fait par rapport aux conditions du réseau, de la charge du système, de la machine (taille de l'écran,). À partir d'un document formaté partiellement statiquement, le système de présentation va appliquer une (ou plusieurs) phases de formatage dynamique pour adapter la présentation aux nouvelles conditions.

3.2 L'avenir du multimédia : de nombreux formats

Nous avons vu qu'il existait de nombreux formats de spécification pour les documents multimédias, il faut aujourd'hui prendre en compte ce paramètre et essayer de fournir des outils auteur capables de manipuler un ensemble de formats. L'émergence de standards comme XML permet d'homogénéiser la manière de décrire les documents et de spécifier les propriétés sous forme d'attributs. Ainsi, les langages SMIL et Madeus sont décrits en XML. L'intérêt d'avoir un moyen standardisé de décrire les données permet de concevoir des outils communs pour décrire la présentation.

Néanmoins, l'émergence de standards comme SMIL 2.0 nécessitera une adaptation des outils auteurs.

SMIL 2.0

Les objectifs de SMIL 2.0 sont, dans un premier temps, d'étendre l'expressivité de SMIL 1.0 et, dans un deuxième temps, de rendre plus modulaire la spécification de SMIL.

Les principales extensions de l'expressivité ont été réalisées dans la définition de la synchronisation temporelle. L'auteur peut maintenant, par exemple, définir des objets maîtres dans les relations de synchronisation. Il peut aussi spécifier de manière plus complète l'interactivité dans le document.

Le deuxième changement important dans SMIL est la spécification de modules. Les principaux modules sont : structure, lien, média, animation, synchronisation,

Ces différents modules permettent de définir la présentation (spatiale, temporelle) ainsi que la structure de navigation d'un document. Un des intérêts est aussi de permettre l'utilisation d'un sous-ensemble de modules dans d'autres contextes. On peut noter que de ce fait, SMIL 2.0 permet de définir un ensemble de langages à partir de sous-ensembles de modules.

On peut noter que ces deux extensions vont dans le sens des travaux menés dans Kaomi.

L'augmentation de l'expressivité dans le module temporel fait que SMIL se rapproche de langages comme MHML. Ainsi, les différentes facilités d'édition fournies par le système auteur restent pertinentes.

On se rapproche donc de documents dont le contenu est décrit dans une syntaxe XML, et dont la présentation est définie grâce à un ensemble de modules SMIL. Ce sous-ensemble évolue en fonction du contexte. Cela nous incite à penser que, dans certains cas, on voudra spécialiser le système auteur pour ce sous-ensemble de modules. L'architecture de la boîte à outils est adaptée à ce besoin.

3.3 Extensions à apporter aux technologies contraintes

L'utilisation de contraintes dans un environnement d'édition permettra à l'utilisateur de paramétrer la résolution. Pour cela il est nécessaire d'étendre l'environnement auteur actuel sur deux points :

- **Politique de formatage adaptable** : comme nous avons pu le voir précédemment, un des points positifs des résolveurs est qu'ils permettent de définir une politique de formatage quel que soit le langage édité. Cependant, cette politique de formatage est dépendante du contexte d'utilisation. Dans certains cas, l'auteur désirera que tous ses objets aient la même durée (présentation de photographies en séquence), et dans d'autres cas, il désirera fixer une durée pour certains objets et laisser le formateur faire au mieux pour le reste, de manière, par exemple, à limiter la bande passante moyenne nécessaire pour visualiser le document. On doit donc fournir à l'auteur le moyen de définir une politique de formatage, ce qui n'est pas simple à faire du fait que le bon paramétrage des poids associés aux contraintes se fait souvent de manière empirique. Il est donc nécessaire dans un premier temps de voir comment les

résolveurs de contraintes peuvent faciliter ce contrôle sur le mécanisme de résolution et dans un deuxième temps d'étudier la façon dont la phase de paramétrisation pourrait être assistée par le système auteur.

- **L'aide au diagnostic** : aujourd'hui, peu de travaux existent sur l'explication de la solution calculée par un résolveur. Cette aide est essentielle si on veut que l'utilisateur puisse paramétrer et configurer le formatage, pour que ce dernier satisfasse au mieux ses besoins. De même, il reste des travaux à effectuer dans le domaine de la compréhension d'un système de contraintes incohérent.

Nous avons vu que l'utilisation des contraintes s'est avérée concluante dans le cadre de documents déterministes. L'intégration de relations spatio-temporelles et de médias indéterministes vont soulever de nouveaux problèmes (facteur d'échelle, médias non formatables statiquement) et introduire de nouveaux besoins (formatage dynamique). Pour répondre à ces besoins de nombreux travaux théoriques visent à étendre l'expressivité des CSP.

- **Dynamique et fuzzy CSP** : des travaux formels ont été réalisés dans le cadre des contraintes pour prendre en compte des comportements imprédictifs. Par exemple, les fuzzy CSP [Verfaillie95] semblent offrir un formalisme intéressant. Dans ce formalisme, chaque contrainte est définie non pas par l'ensemble des valeurs qui la satisfont mais par un ensemble flou : l'ensemble des valeurs qui la satisfont plus ou moins. Dans ce cadre, une contrainte est satisfaite à un certain degré plutôt que satisfaite ou insatisfaite. La richesse de ce modèle permet de prendre en compte à la fois des préférences sur les valeurs et des priorités sur les contraintes. Dans ce modèle, l'acceptabilité globale d'une solution est graduelle. Les meilleures solutions sont celles qui maximisent le degré de satisfaction de la moins satisfaite des contraintes. C'est-à-dire, que si nous définissons le coefficient de qualité d'une solution comme la valeur du minimum des degrés de satisfaction des contraintes composant le problème, alors une bonne solution est celle qui a le coefficient de qualité le plus important. La complexité est du même ordre de grandeur que pour les CSP. Cette richesse permet d'exprimer les différentes préférences nécessaires à la résolution du système de contraintes lors de la phase d'édition.
- **Contraintes multidimensionnelles** : l'introduction de ces nouveaux besoins d'expression font augmenter sensiblement le nombre de contraintes. Actuellement nous avons un système de contraintes pour chacune des dimensions (temporelle, spatiale), nous pourrions en avoir un pour calculer, par exemple, les polices de caractères, la taille de ces polices, la couleur, Cependant, si on spécifie des dépendances entre ces différents éléments, les graphes de contraintes obtenus sont trop complexes pour les solveurs actuels. Une théorie a été développée pour prendre en compte de tels problèmes [Verfaillie95], cependant peu de solveurs prennent en compte ces particularités.