GRENOBLE I
UNIVERSITE
JOSEPH FOURIER
SCIENCES. TECHNOLOGIE. MEDECINE

Université Joseph Fourier
U.F.R
Informatique & Mathématiques Appliquées

I.M.A.G

Magistère Informatique :

Projet présenté par:

Laurent TARDIF

Traitement et visualisation de la flexibilité d'un scénario temporel

Effectué dans le projet Opéra – INRIA

Date:
Septembre 1997

Membres du Jury:
Muriel Jourdan
Claude Puech

Remerciements

Je remercie tout d'abord Claude Puech pour avoir bien voulu faire partie de ce jury.

Je remercie Cécile ROISIN et Vincent QUINT pour m'avoir accueilli au sein du projet Opéra où j'ai effectué mon stage.

Je tiens à exprimer ma profonde gratitude à Muriel JOURDAN responsable de mon stage pour m'avoir accueilli et pour son dévouement constant à m'apporter l'aide dont j'avais besoin et les remarques qu'elle a apportées à mon travail.

Je remercie chaleureusement Nabil LAYAÏDA, sans qui ce travail n'aurait jamais pu avoir lieu. Loay Sabry, qui visiblement reprend forme :–)

Je remercie Cécile ROISIN pour sa patience lors de la relecture de mon rapport.

Je remercie enfin José, Christophe, Frédéric, Laurent, Cédric, toute l'équipe d'Opéra et les membres du W3C pour leur sympathie.

Introduction

Chapitre I

Visualisation du scénario temporel dans Madeus

Chapitre II

Visualisation et manipulation de la flexibilité et des opérateurs d'interruption

Chapitre III

Contraintes

Chapitre IV

Mise en œuvre

Conclusion

Introduction

1 Présentation générale

Aujourd'hui, les documents électroniques intègrent une dimension temporelle en plus des dimensions spatiale (présentation graphique du document), logique (décomposition en parties,...) et sémantique (hypertexte). L'édition de documents comportant des objets multimédia (vidéo, audio, images, etc) organisés dans la dimension temporelle est une tâche complexe. En effet, la nature dynamique des objets manipulés (tels que la vidéo et l'audio) ainsi que la définition de leur organisation temporelle empêche la fusion des phases d'édition et de présentation. Le principe statique du WYSIWYG (What You See Is What You Get) fort utile lors de l'édition d'un document classique ne peut donc s'appliquer à la création d'un document multimédia.

Edition de documents multimédia

De ce fait, les outils d'édition actuels de ce type de documents ressemblent plus à des outils de programmation, par l'utilisation de langages de script pour décrire l'organisation temporelle des objets. Néanmoins, il serait dommage de limiter les auteurs potentiels de documents multimédia aux seuls experts informaticiens. Un des objectifs fixés par le projet Opéra est la création d'un éditeur multimédia puissant ayant une interface graphique conviviale, utilisable par des auteurs non–informaticiens.

La seule dimension d'un document multimédia qui nous intéresse dans ce mémoire est la dimension temporelle dans laquelle est définie la façon dont s'organisent les objets dans le temps(1). On appelle **scénario temporel** d'un document multimédia la description de cette organisation. La description de ce scénario (textuelle, graphique) fait partie de la **phase d'édition** du document. Lors de la **phase de présentation** les objets sont placés dans le temps de manière à respecter les consignes du scénario temporel.

Un système auteur pour la composition de documents multimédia se compose de deux sous–systèmes : le système d'édition et le système de présentation. Le rôle du **système d'édition** consiste à alléger la charge de

(1) Les autres domaines ont fait l'objet d'important travaux depuis de nombreuse années ([17],[[?]]), contrairement à la dimension temporelle plus récemment pris en compte.

l'auteur lors de la phase d'édition (interface d'édition, détection des erreurs, aide à la conception, etc.) tandis que le système de présentation a pour fonction d'orchestrer l'exécution du document (affichage, déplacement, démarrage et interruption des objets, etc.). Madeus, nom donné au prototype développé au sein du projet Opéra, est un système conçu selon ce principe.

Edition de documents multimédia avec le prototype Madeus

L'approche choisie dans Madeus[31] pour la description du scénario temporel d'un document multimédia est fondée sur l'utilisation de contraintes qui expriment le placement temporel relatif des objets. L'idée sous–jacente est qu'un auteur puisse dire : "Cette vidéo A se joue avant cette autre vidéo B. Un texte C se joue en même temps que la présentation de la vidéo B. La Vidéo A démarre en même temps qu'un son D et D doit forcément être terminé lorsque C démarre". Ce qui se traduit par les contraintes suivantes:

- `Vidéo A avant Vidéo B`
- `Texte C égale Vidéo B`
- `Vidéo A démarre Audio D`
- `Audio D avant Texte C`

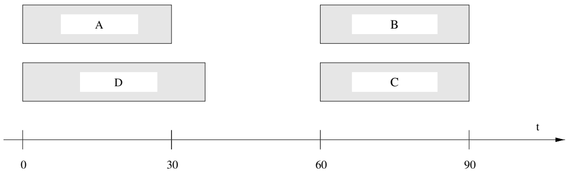
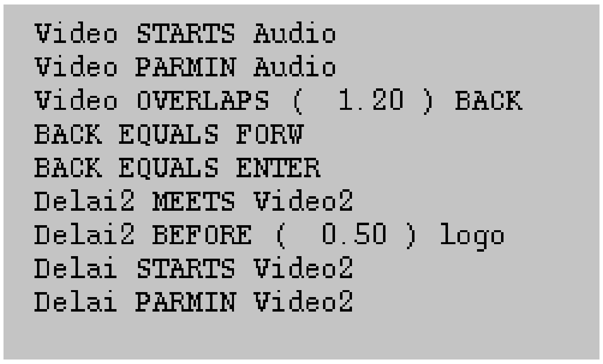


Figure I.1 : Une exécution possible du scénario précédent

La Figure I.1 présente un placement possible des objets A, B, C et D à partir de ce scénario en supposant que leur durée respective est 30s, 30s, 30s et 40s.

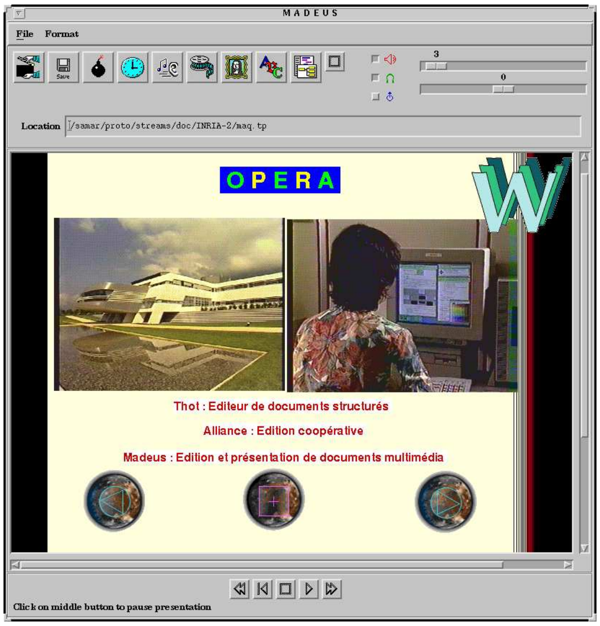
La spécification du scénario avec des contraintes souples comme *avant, après, pendant* permet la définition de plusieurs placements temporels à partir d'un seul scénario. En effet, lorsque l'auteur spécifie "A avant B", il ne précise aucune valeur pour l'intervalle entre la fin de A et le début de B. C'est le système lors de la présentation du document qui effectuera le **placement temporel global** des objets en fonction des **informations locales** (contraintes entre deux objets ou groupes d'objets). Le rôle de l'auteur est donc de spécifier uniquement les informations locales, laissant au système la tâche fastidieuse de mettre à jour les informations globales.

Cette approche de spécification du scénario temporel à base de contraintes est aussi celle choisie par d'autres projets tels que ISIS [45] et HPAS [30].



```
Video STARTS Audio
Video PARMIN Audio
Video OVERLAPS (  1.20 ) BACK
BACK EQUALS FORW
BACK EQUALS ENTER
Delai2 MEETS Video2
Delai2 BEFORE (  0.50 ) logo
Delai STARTS Video2
Delai PARMIN Video2
```

Figure I.2 : Le scénario sous forme textuelle



Location /samar/proto/streams/doc/INRIA-2/maq.tp

Thot : Editeur de documents structurés

Alliance : Edition coopérative

Madeus : Edition et présentation de documents multimédia

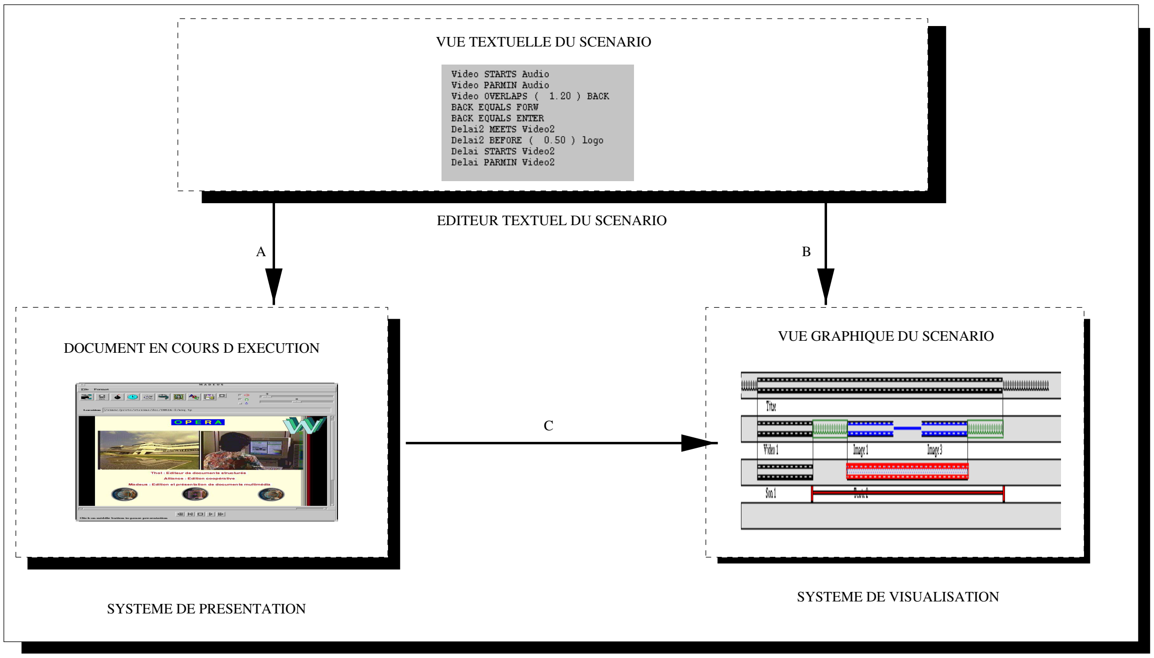
Click on middle button to pause presentation

Figure I.3 : Interface de présentation de Madeus

Actuellement dans Madeus il est possible de spécifier textuellement le scénario temporel d'un document comme sur la Figure I.2. A partir de cette spécification textuelle le système est capable de présenter le document en positionnant les objets dans le temps de manière à respecter les contraintes du scénario. La Figure I.3 présente l'interface de présentation de Madeus.

Afin de faciliter la tâche de l'auteur lors de la construction du document, nous nous nous intéressons à la **visualisation de l'ensemble des placements temporels possibles des objets** à partir d'un scénario.

Par la suite nous appellerons la vue ainsi obtenue, **la vue graphique du scénario.** Il ne s'agit donc pas simplement d'offrir un équivalent graphique à la forme textuelle du scénario, comme cela est fait dans les autres systèmes fondés sur une approche à base de contraintes, mais de restituer à l'auteur des informations globales déduites du scénario. En reprenant l'exemple de la Figure I.1, on cherche à concevoir une vue qui permette à l'auteur de visualiser les quatre objets A, B, C et D sur des axes temporels avec la possibilité de déplacer ces objets sur ces axes, en fonction des contraintes qui les régissent, de manière à pouvoir appréhender les différents placements possibles.



```
Video STARTS Audio
Video PARMIN Audio
Video OVERLAPS (  1.20 ) BACK
BACK EQUALS FORW
BACK EQUALS ENTER
Delai2 MEETS Video2
Delai2 BEFORE (  0.50 ) logo
Delai STARTS Video2
Delai PARMIN Video2
```

Figure I.4 : Processus de création souhaité dans Madeus

Ainsi, le processus de création d'un document multimédia au sein de Madeus doit être étendu pour intégrer la vue graphique du scénario temporel que nous venons de présenter (Figure I.4). L'auteur pourra s'aider du système de visualisation (flèche B de la Figure I.4) pour affiner son scénario avant de le présenter (flèche A de la Figure I.4) et éventuellement le modifier. Il pourra s'aider de la vue graphique pour comprendre cette exécution (flèche C de la Figure I.4). Ces trois vues, étant une représentation de la même information, il est nécessaire de maintenir et d'assurer la cohérence entre les vues.

2 Objectifs du stage

Un premier travail lié à la visualisation du scénario temporel a été effectué au cours de mon projet de DEA [47]. Il a conduit à la définition d'un principe de base qui a été mis en œuvre dans un prototype pour un sous ensemble de scénarios potentiellement exprimables. Dans le cadre du magistère le travail consistera à étendre la solution proposée au cours du DEA de manière à intégrer deux notions particulièrement importantes dans Madeus :

- la flexibilité des objets, c'est à dire leur capacité à être définis par un intervalle de durée possible et non pas par une valeur unique ;
- les opérateurs d'interruption qui permettent d'exprimer des scénarios plus complexes que ceux considérés durant le DEA.

3 Démarche de travail

Dans une première partie nous présenterons Madeus ainsi que le système de visualisation du scénario temporel que nous avons réalisé au cours du DEA.

Dans une seconde partie nous présenterons les difficultés liées à la manipulation d'objet flexibles et des opérateurs d'interruption. Nous verrons que cette intégration nécessite une bonne connaissance des méthodes actuelles de résolution de contraintes et de maintien de solutions dans des systèmes à base de contraintes. Nous présenterons différentes approches dans la troisième partie. Finalement dans la quatrième partie nous présenterons la solution adoptée.

Chapitre I

Visualisation du scénario temporel dans Madeus

I.1 Les approches à base de contraintes, Madeus

Une idée commune aux approches de spécification du scénario temporel d'un document multimédia fondées sur l'utilisation de contraintes est de permettre à l'auteur de ne pas fixer précisément les durées de chaque objet ainsi que leur date de démarrage, mais de laisser le système les calculer en fonction des contraintes exprimées par l'auteur.

Pour cela, chaque objet multimédia est caractérisé par trois valeurs:

- durée minimum ;
- durée idéale ;
- durée maximum.

C'est un moyen de prendre en compte la flexibilité des objets: leur capacité à être présentés avec des durées variables. Par exemple, un auteur qui tient à présenter une image durant la présentation d'une vidéo qui doit elle même être présentée en même temps qu'une image, déclare dans son scénario les contraintes de placement entre les objets. Le système calcule le temps de présentation de chacun de manière à respecter les contraintes. On parle de "formatage temporel" par analogie au formatage spatial.

Un des problèmes qui apparaît dans ces approches est la notion de cohérence du scénario temporel qui caractérise le fait qu'il existe au moins une solution de placement. Ce problème ne fait pas partie des préoccupations du stage présenté dans ce rapport.

Les contraintes portent soit sur les instants de début et fin, c'est le cas par exemple de FireFly [31] où l'auteur indique que deux instants doivent être simultanés ou en séquence, soit sur les objets, c'est le cas d'ISIS [17]et de

Madeus [19] où l'auteur indique que deux objets doivent être joués en même temps, l'un après l'autre ...

I.1.1 Contraintes d'intervalles: l'exemple de Madeus

Madeus est un prototype développé au sein du projet opéra [19]. Les particularités de l'approche choisie dans Madeus sont:

- le typage des objets ;
- l'extension des opérateurs d'Allen ;
- la décomposition logique du document ;
- l'utilisation de plusieurs type de liens.

Ce sont à présent ces quatre points qui vont être à présent détaillés.

- La différenciation de deux types d'objets:
  - Objets contrôlables: leur durée de présentation peut être librement fixée par le système entre deux bornes. C'est le cas par exemple des objets de type texte. On peut leur associer une borne minimale pour assurer la lisibilité de leur contenu et une borne maximale infinie.
  - Objets incontrôlables: leur durée de présentation ne sera connue qu'à l'exécution du scénario car: soit elle dépend de facteurs dynamiques comme la charge de la machine ou du réseau (cas d'une vidéo), soit elle est par nature indéterministe comme une interaction utilisateur. On peut dans le premier cas donner des bornes minimales et maximales sur la durée en considérant des critères de qualité de présentation des objets.
- L'utilisation des opérateurs d'Allen ([1]) comme jeu de contraintes augmentés de deux opérateurs d'interruption supplémentaires (le parmin et le parmaster). Allen a défini l'ensemble de toutes les positions relatives que peuvent avoir deux intervalles placés sur un axe temporel. Nous en présentons sept dans la Figure 1.1, les six autres s'obtiennent en prenant les placements symétriques (le symétrique de l'opérateur "equal" est lui–même). Madeus utilise les opérateurs d'Allen comme autant de contraintes que souhaite voir respecter l'auteur du scénario. Il peut en plus quantifier les délais libres introduits par certains opérateurs (before, during, overlpas qui deviennent before(t), during(t), overlaps(t)):

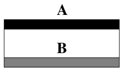
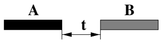
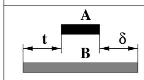
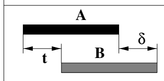
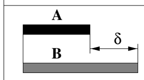
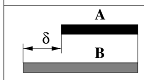
Relations	Sémantique Graphique	Contraintes Numériques
A equals B		dur(A) = dur(B)
A before B		t > 0
A during B		t < dur(B) - dur(A) & t > 0 & δ > 0
A overlaps B		t > dur(A) - dur(B) & t > 0 & δ > 0
A meets B		——
A starts B		dur(A) < dur(B)
A finishes B		dur(A) < dur(B)

Figure 1.1 : Les opérateurs d'Allen

- La définition d'une structure hiérarchique du document qui n'est pas le support de la composition temporelle (ie: ce n'est pas un arbre d'opérateurs). Par exemple, sur la hiérarchie de la Figure 1.2, il est possible de placer des opérateurs entre les 3 scènes ou entre les objets Son3, Vidéo4, et le groupe composite, mais il est interdit de placer directement des contraintes entre Photo1 et Son3 ou entre Son1 et Son2.

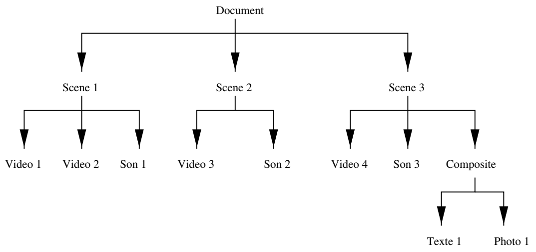


Figure 1.2 : Exemple de hiérarchie d'un document multimédia

La création d'un document multimédia se fait par la création de la structure hiérarchique et la définition de contraintes entre les objets. A tout moment l'auteur peut ajouter un objet dans la hiérarchie et/ou ajouter des contraintes.

- La modélisation des liens hypertextes par des attributs associés aux objets du document multimédia. Ces attributs contiennent le type du lien (inclusion ou référence) et la destination qui doit être un objet composite.

Par exemple l'objet Photo 1 peut avoir un attribut du type référence qui pointe sur scène 1. Cela signifie que lors de la présentation du document, le texte 1 sera une zone activable : si on clique sur texte 1, la scène 3 est arrêtée et la scène 1 est lancée.

C'est liens peuvent être de différentes natures :

- Inclusion, dans ce cas lorsque le lien est activé, un objet (qui peut être un objet composite) est inséré dans le document en cours.
- Référence, dans ce cas lorsque le lien est activé le document en cours est stoppé, et on présente l'objet désigné par le lien. Cet objet peut être de deux types:
  - Objet du document, dans ce cas on présente un objet du document courant.
  - Document à l'extérieur du document courant, cela permet la navigation entre les documents.

I.2 Visualisation du scénario temporel dans Madeus

Dans le cadre du DEA [28] nous avons proposé une première solution partielle pour répondre au besoin de visualisation. Cette solution ne prenait en compte ni la flexibilité des objets, ni les opérateurs d'interruptions. Le principe de cette solution est le suivant : visualiser un placement des objets qui respecte les contraintes du scénario, et donner la possibilité à l'auteur de déplacer les objets dans la limite autorisée par le scénario, afin de parcourir l'ensemble des placements possibles. Par conséquent la seule flexibilité prise en compte dans ce travail provient des relations qui comme le `Before` ou le `During` introduisent des délais pouvant prendre différentes valeurs. Nous allons à présent détailler cette première solution.

I.2.1 Visualisation statique

Choix de visualisation

Les objets de base sont représentés par des rectangles dont la longueur est proportionnelle à leur durée effective. Chaque type d'objet (vidéo, son, texte, image) est représenté par un motif discriminant. Les objets sont placés sur des lignes horizontales qui représentent des axes temporels, le temps s'écoulant de gauche à droite. Les lignes donnent aussi l'intuition du déplacement temporel par un placement graphique horizontal des objets).

Les contraintes introduisent trois types d'éléments dans la vue scénario :

- Les traits horizontaux qui modélisent les délais fixes (`Image1 before (15) Image2`).
- les ressorts qui modélisent les délais flexibles (`Video1 Before Image1`).
- Les traits verticaux, qui représentent la contrainte de simultanéité de deux instants (`Image1 Start Texte2`).

Le placement choisi essaie de minimiser la visualisation de fausses informations (pas d'alignement d'objets s'il n'est pas imposé par le jeu de contraintes) et la superposition des objets et des relations lors du placement vertical des objets.

La représentation explicite des objets, des contraintes et des délais introduits par ces contraintes permet une interprétation directe du scénario sans nécessairement avoir recours à la vue textuelle.

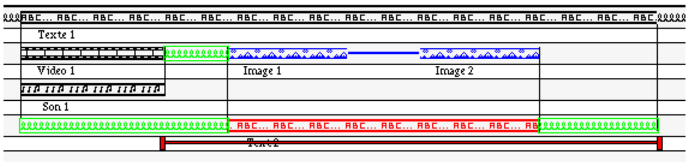


Figure 1.3 : Visualisation du scénario temporel

Interaction de l'utilisateur

Afin d'aider l'auteur dans le parcours de l'ensemble des solutions, nous visualisons l'intervalle de déplacement de manière à ce qu'il puisse anticiper le comportement de l'objet qu'il manipule (critère de prévisibilité). Dans l'exemple de la Figure 1.3 l'auteur a sélectionné l'objet `Texte2`. Un trait rouge apparaît sous la ligne où est placé l'objet sélectionné, indiquant ainsi sa latitude de déplacement. Sur notre exemple, `Texte2` est limité à sa gauche par la fin de `Vidéo1` et à sa droite par la fin de `Texte1`.

Les objets qui sont reliés à l'objet sélectionné par des contraintes rigides sont colorés en bleu. Ces objets sont solidaires de l'objet sélectionné dans ses déplacements. Sur notre exemple, c'est le cas des objets `Image1` et `Image2`. Les intervalles qui jouent le rôle d'amortisseur (entre les objets fixes et les objets qui bougent) sont colorés en vert. C'est le cas des trois délais flexibles de notre exemple.

Liaison entre la vue scénario et la vue textuelle

Il est nécessaire de faire explicitement le lien entre les différentes vues du document. Le premier service consiste à offrir une certaine forme de synchronisation entre la vue scénario et la vue textuelle par une sélection simultanée dans les deux vues des éléments concernés.

I.2.2 Visualisation dynamique

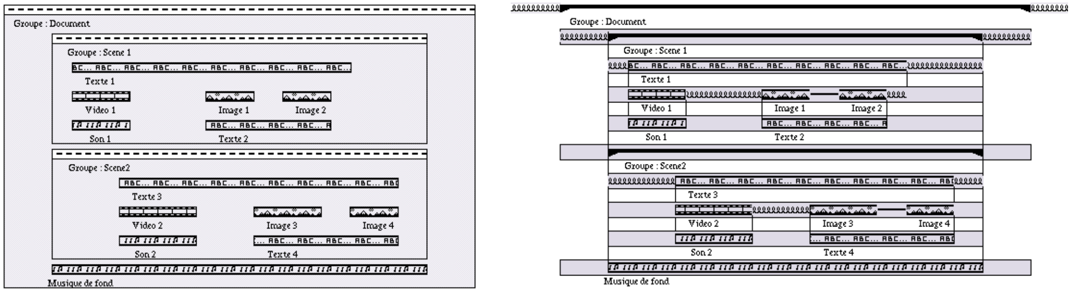
L'idée majeure de la visualisation dynamique du scénario est d'offrir la possibilité à l'auteur de **manipuler la vue scénario** par déplacement des objets représentés dans cette vue. L'objectif est de permettre à l'auteur d'accéder à l'espace de solutions du scénario, grâce aux actions qu'il effectue sur cette vue.

L'auteur peut déplacer l'objet sélectionné dans tout son intervalle de validité, le système de visualisation mettant à jour au fur et à mesure du déplacement la représentation de la solution correspondante. Il est intéressant de noter que les limites de ce déplacement correspondent toujours à l'absorption complète d'un délai flexible du scénario. Par conséquent, et c'est là un des avantages de notre solution, l'auteur peut comprendre facilement la raison d'un blocage. En effet, grâce au mécanisme de liaison avec la vue textuelle, il peut en cliquant sur le délai en question connaître la relation qui l'a engendré.

I.2.3 Visualisation des dimensions logique et temporelle

Le besoin d'offrir à l'auteur le moyen de percevoir l'enchaînement temporel d'objets de composants différents rend complexe le problème d'intégration de la hiérarchie, puisque toute méthode de représentation de la hiérarchie (arbre, structure indentée, arbre conique) est non satisfaisante.

Nous avons donc opté pour un affichage de la hiérarchie à l'intérieur de la vue du scénario temporel tout en permettant à l'auteur de se focaliser sur la structure hiérarchique par une méthode de filtre, activé à la demande, qui supprime les informations temporelles (partie gauche de la Figure 1.4). Sur la partie droite de cette figure les mêmes composants (scène1, scène2, musique de fond) sont représentés avec leur contenu et les relations temporelles, ce qui permet à l’auteur de percevoir le placement dans le temps de tous ces objets). Sur la partie droite de cette figure les mêmes composants (scène1, scène2, musique de fond) sont représentés avec leur contenu et les relations temporelles, ce qui permet à l’auteur de percevoir le placement dans le temps de tous ces objets.



Vue obtenue par filtrage **Vue complète**

Figure 1.4 : Visualisation temporelle et hiérarchique

I.3 Conclusion

Au cours de ce chapitre nous avons présenté l’approche prise dans Madeus pour l’édition d’un document multimédia, ainsi qu’une première réponse aux problèmes de visualisation graphique du scénario temporel. Celle−ci a permis de dégager un principe de base en considérant uniquement un sous ensemble des scénarios exprimables dans Madeus. Deux points majeurs n’y sont pas intégrés: la flexibilité des objets et les opérateurs d’interruption. L’intégration de ces deux points fait l’objet du cœur de ce mémoire et des chapitres suivants.

Chapitre II

Visualisation et manipulation de la flexibilité et des opérateurs d'interruption

Dans un premier temps nous allons identifier les besoins en visualisation liés à l'intégration d'objets flexibles dans un scénario et dans un deuxième temps nous présenterons les besoins et les difficultés liés à la représentation et à la manipulation des opérateurs d'interruption.

II.1 Flexibilité des objets

II.1.1 Introduction

Lors de la visualisation d'une solution d'un scénario qui contient des objets flexibles l'auteur peut parcourir l'ensemble des solutions soit :

- en déplaçant les objets ;
- en modifiant la durée des objets flexibles.

En réaction à ces modifications, le système de visualisation est amené à déplacer d'autres objets et/ou modifier leur taille pour pouvoir présenter à l'auteur un placement cohérent par rapport au scénario initial.

Il est donc important de distinguer les objets flexibles de manière à ce que l'auteur sache quels sont les objets dont la durée est susceptible d'être modifiée. Une idée intuitive consiste à représenter la flexibilité par une métaphore explicite sur la représentation graphique de l'objet (ex: un ressort). Cependant, cette information est loin d'être suffisante et ce sont tous les besoins liés à l'utilisation d'objets flexibles dans les scénarios qui vont être à présent identifiés. Ceux–ci sont regroupés en trois catégories :

- Anticipation, ce sont les besoins liés à la phase qui précède la modification, c'est à dire l'intervalle de modification possible.
- Mode d'action, ce sont les besoins liés aux modes d'actions et au actions de l'auteur.
- Réponse, ce sont les besoins liés à la façon dont le système va réagir à la manipulation de l'auteur.

II.1.2 Besoins liés à la manipulation des objets flexibles

II.1.2.1 Anticipation du comportement du scénario

Savoir qu'un objet est flexible dans un scénario donné ne constitue pas une information suffisante pour anticiper les conséquences d'une manipulation (déplacement ou retaillage). Il est important que l'auteur connaisse avant la modification un certain nombre d'informations qui vont lui éviter d'avoir à effectuer de nombreuses manipulations de la vue graphique afin de mieux comprendre l'ensemble des solutions associées à son scénario.

- **Connaissance et anticipation de l'intervalle de flexibilité.**
    - **Degré de flexibilité de la durée d'un objet**

      Il faut représenter l'intervalle de flexibilité associé à la durée de l'objet dans le scénario (flexibilité qui est différente de la flexibilité intrinsèque de l'objet). Par exemple, un objet ayant une flexibilité intrinsèque définie entre 4 et 9 peut, dans un scénario donné (à cause des contraintes définies dans le scénario) voir sa flexibilité réduite à l'intervalle [5..8]. Cet intervalle signifie que pour chacune de ses valeurs, il existe une durée et un placement des objets tel que le scénario obtenu est cohérent. Par conséquent cette information permet à l'auteur de savoir avant toute manipulation dans quelle mesure un objet peut être retaillé tout en conservant l'existence d'une solution cohérente à son scénario.
    - **Anticipation du degré de flexibilité à droite et à gauche**

      Il faut permettre à l'auteur d'anticiper l'intervalle de flexibilité dont il dispose lors d'un retaillage par le bord gauche ou le bord droit d'un objet au moment opportun. Le système devra donc offrir à l'auteur une visualisation de cette flexibilité, par exemple à l'aide d'une visualisation de cet intervalle par une représentation graphique au dessus de la représentation de l'objet sélectionné.

Dans l'exemple de la Figure 2.1 l'objet B a une durée de 5, mais celle–ci peut varier de 4 à 7 (c'est l'intervalle de flexibilité de sa durée). Dans le premier cas l'objet peut être agrandi de deux et diminué de un par la gauche et par la droite.

Dans le second cas, il ne pourra se retailler que par la droite en considérant que A et D ne sont pas flexibles.

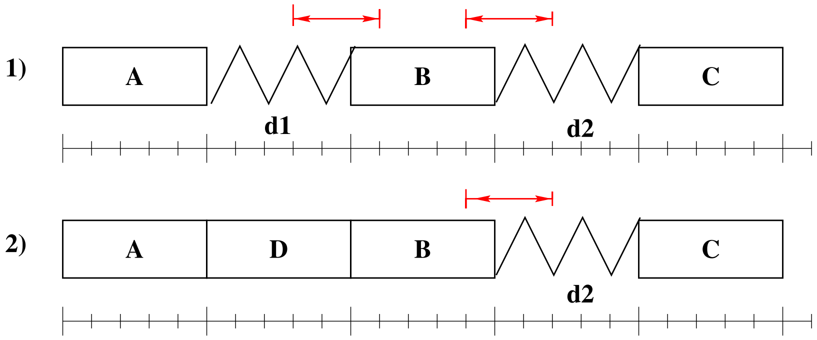


Figure 2.1 : Contrôle de la flexibilité

- **Visualisation des objets affectés**

  Une fois que l'auteur a choisi un objet sur lequel va porter ses modifications, il faut lui permettre d'anticiper le comportement de son scénario, en visualisant par exemple les objets qui vont être modifiés (déplacés ou retaillés) lors de ses manipulations.

II.1.2.2 Mode d'action

Il est évident qu'il faut permettre à l'auteur de modifier interactivement la taille des objets. Pour cela il faut lui offrir des moyens de manipuler leur taille, que ce soit au moyen d'une boîte de dialogue et/ou par une manipulation directe à l'aide de la souris sur les bords de la représentation graphique de l'objet. En fait ces deux modes d'interaction sont complémentaires, car la manipulation à la souris est intuitive, et la manipulation par des boîtes de dialogue permet de réaliser des ajustements précis.

II.1.2.3 Réponse du système: préventive, locale, continue et pertinente

- **Interdire les manipulations amenant à des scénarios incohérents.**

  Le système doit assurer que chaque solution visualisée est cohérente. Pour cela, il doit par exemple empêcher de déplacer ou retailler des objets si ces déplacements amènent à des incohérences, cela peut être fait en bloquant les objets quand cela est nécessaire. Il faut aussi offrir les moyens à l'auteur de comprendre le moment où son objet sera bloqué.

- **Continuité des déformations**

  Pour les mêmes raisons (faciliter la tâche de compréhension de l'auteur) il est souhaitable que les déformations se fassent de manière continue. C'est à dire que le système doit s'adapter en continu aux manipulations de l'auteur et il ne doit pas attendre la fin de la manipulation de l'auteur pour s'adapter.

- **Multiplicité des réponses**

  Dans le cas où le système permet à l'auteur de retailler ou de déplacer les objets, il se peut que plusieurs choix soient offerts au système pour répondre à ces déformations.

  Dans l'exemple de la Figure 2.2, si l'auteur a décidé de retailler l'objet B par la droite, le système a le choix entre :

  - réduire le délai d2 (cas 2)
  - réduire l'objet C (si celui ci est flexible) (cas 3)
  - répercuter l'amortissement sur les objets délai2 et C (cas 4).

  Quelle que soit la solution choisie, il ne faut en visualiser qu'une et permettre à l'auteur d'accéder aux autres (par exemple par manipulation de la solution présentée).

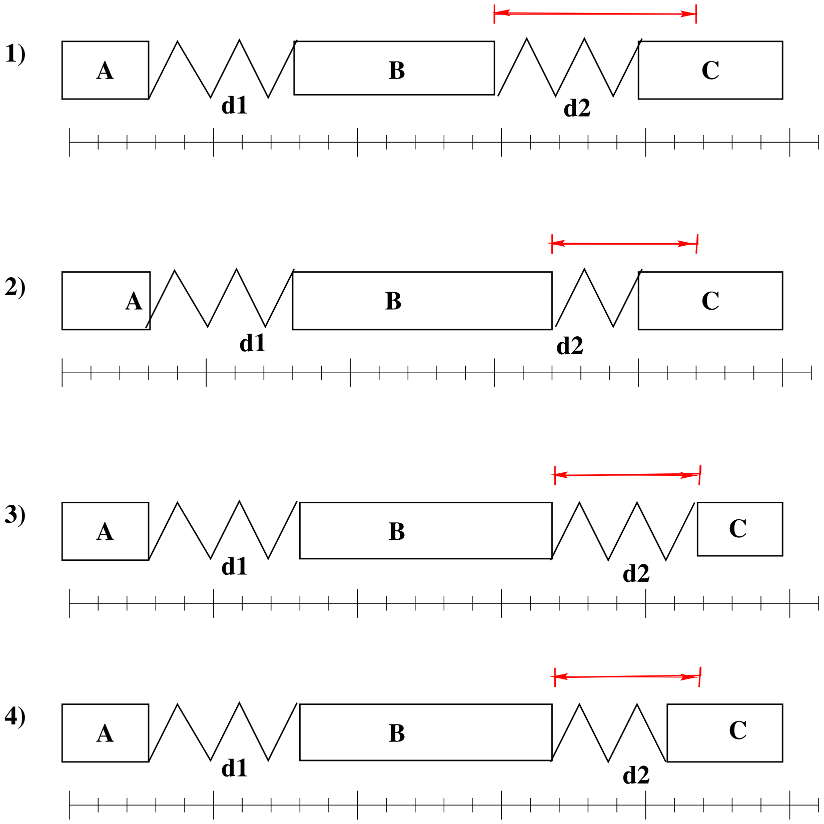


Figure 2.2 : Contrôle de la flexibilité

Le système de visualisation doit pouvoir contrôler la réponse à une déformation de manière à satisfaire les deux critères suivants :

- préférence dans le choix des objets à déformer, on préférera en effet toujours retailler un délai introduit par le système qu'un objet introduit par l'auteur ;
- localité des objets déformés par rapport à l'objet sélectionné. Il est important d'offrir à l'auteur plusieurs limites à ses déformations (déformations plus ou moins locales). En effet, pour qu'il puisse garder le contrôle sur les déformations, il faut que celles–ci se limitent à un niveau local, mais pour que cela ne restreigne pas l'espace de solutions possibles de son document, il est nécessaire de permettre des déformations globales.

II.1.3 Ebauche de solution pour la réponse du système

Dans l'exemple de la Figure 2.3 on peut dire que la déformation des délais d1 et d2, en réponse à une manipulation sur l'objet B, est plus locale que la déformation d'un des objets de la Scène 2. De plus elle privilégie la déformation d'un délai ce qui correspond plus au résultat attendu. C'est cette solution qui nous parait la plus pertinente dans ce cas précis.

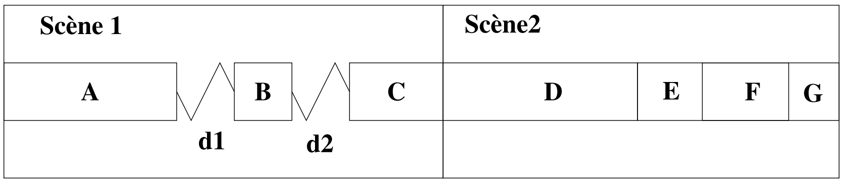


Figure 2.3 : Localité des déformations

De manière générale, on essaye en premier lieu de ne perturber que le niveau courant. Dans le niveau courant on essaye d'abord de trouver des délais qui peuvent amortir les déformations de l'auteur. Si plusieurs choix s'offrent à nous, on choisit de déformer les objets qui sont les plus proche de l'objet manipulé. Si les délais ne suffisent pas à amortir les déformations de l'auteur on les utilisent au maximum, et on utilise la flexibilité des objets les plus proches de l'objet déformé.

S'il n'y a pas de possibilité au niveau courant et que l'auteur le demande explicitement, on essaye de faire la même chose au niveau supérieur et ainsi de suite jusqu'à trouver une solution.

II.2 Les opérateurs d'interruption: Besoins et difficultés

II.2.1 Définition

Les opérateurs d'interruption ont pour effet de provoquer la terminaison des objets liés par l'opérateur et des objets liés (par transitivité) à l'objet interrompu. Les opérateurs d'interruption définis dans Madeus sont :

- A $\text{parmaster}_{(A)}$ B, qui est équivalent à A Parmin B si A est plus court que B, ou à B start A si A est plus long que B. Par conséquent la durée de B est limitée par celle de A.
- A parmin B, qui exprime l'interruption de l'objet le plus long par l'objet le plus court ;

Nous pouvons voir graphiquement dans la Figure 2.4 la sémantique de ces opérateurs.

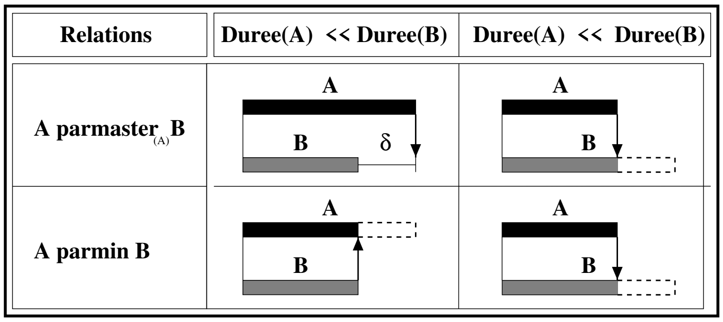


Figure 2.4 : Les opérateurs d'interruption

Lorsque la durée des objets concernés par les opérateurs d'interruption est unique, il est facile pour le système de déterminer quels sont les objets qui sont interrompus. Par exemple dans la Figure 2.5 où nous avons deux objets A et B dont la durée peut varier entre 3 et 4, trois cas sont possibles :

- Dans le cas où durée(A) < durée(B), l'objet B est interrompu et potentiellement ne délivre pas la totalité de son contenu sémantique.
- Dans le cas où durée(A) = durée(B), aucun des deux objets n'est interrompu.
- Dans le cas où durée(A) > durée(B), l'objet A est interrompu et potentiellement ne délivre pas la totalité de son contenu sémantique.

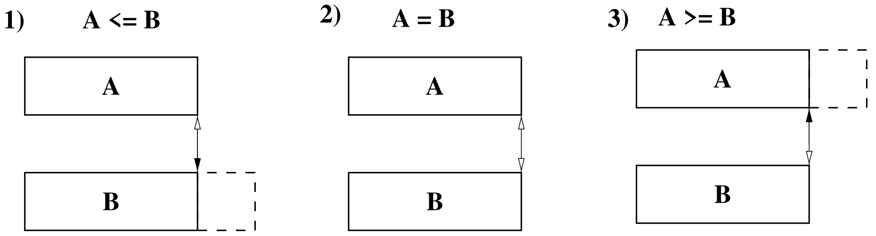


Figure 2.5 : Causalité et flexibilité

II.2.2 Visualisation de l'interruption (sémantique de la coupure)

Il est important pour l'auteur de savoir précisément quels sont les objets qui risquent d'être interrompus, car ceux–ci peuvent potentiellement ne pas délivrer la totalité de leur contenu.

Ambiguïté entre les opérateurs (Equal, Parmin)

Il est nécessaire de bien distinguer l'opérateur parmin de l'opérateur equal. En effet, les deux opérateurs se traduisent par un placement temporel identique : les opérandes démarrent en même temps et se terminent en même temps mais leur sémantique est différente: dans le cas du parmin, il se peut que l'un des deux opérandes ait été interrompu. Cette différence doit apparaître dans la vue graphique du scénario.

Par exemple, dans la Figure 2.6 il est important de montrer que dans le cas où l'auteur a défini la relation `A Parmin B` les deux objets peuvent être potentiellement interrompus.

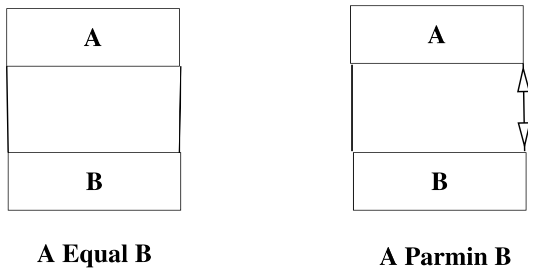


Figure 2.6 : Exemple d'ambiguïté

Propagation de l'interruption

Comme nous l'avons dit précédemment les opérateurs d'interruption ont un effet qui se propage par transitivité sur les objets. Il est donc important de visualiser cette interruption à la fois sur l'objet lié directement mais aussi sur les objets liés par transitivité.

Par exemple, dans la Figure 2.8 on visualise la propagation de l'interruption dans le cas de l'opérateur Equal (Cas 1) et de l'opérateur During (cas 2).

Cas 1 :	Cas 2:
A Parmin B	A Parmin B
C Equal B	C During B

Figure 2.7 : Scénario

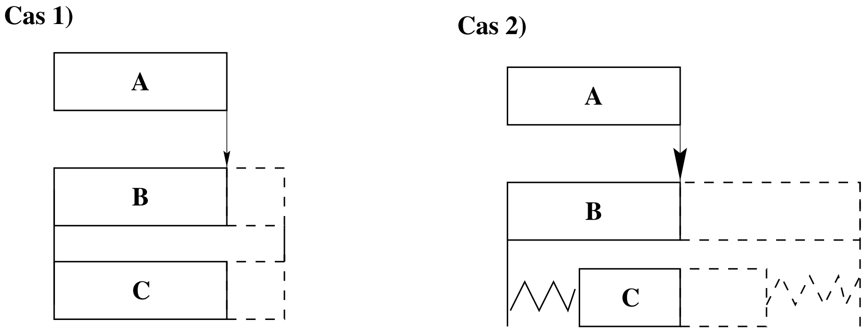


Figure 2.8 : Propagation de l'interruption

II.2.3 Flexibilité et opérateurs d'interruption

Lorsque la durée des objets concernés par les opérateurs d'interruption est unique, il est facile pour le système de déterminer quels sont les objets qui sont interrompus, cependant la combinaison de la flexibilité des objets et des opérateurs d'interruption pose des problèmes nouveaux. Les trois configurations présentées dans la section peuvent apparaître successivement dans le même scénario en fonction de la durée des objets A et B.

Dans les trois cas il faut suggérer à l'auteur l'existence des deux autres et lui permettre d'y accéder. En effet, dans le cas d'objets incontrôlables, l'auteur ne pourra pas savoir a priori lequel des trois cas se produira à l'exécution.

II.3 Conclusion

Comme nous venons de le voir, nous avons plusieurs besoins liés a l'intégration des opérateurs d'interruption et de la flexibilité des objets. Il est important d'offrir à l'auteur une représentation graphique qui lui permette de bien appréhender le comportement de son scénario.

Besoins liés à la visualisation d'information :

1. anticipation des comportements ;
2. non ambiguïté de la représentation ;
3. représentation d'informations multiples sans surcharge cognitive pour l'auteur ;
4. manipulation de la représentation graphique ;
5. contrôlabilité des déformations.

Les besoins identifiés nécessitent des mécanismes de manipulation de contraintes avancés notamment pour le maintien de la cohérence de la solution visualisée. En effet, comme nous avons pu le dire précédemment, à chaque interaction de l'utilisateur sur la représentation graphique du scénario le système devra apporter les modifications nécessaires au scénario pour tenir compte des manipulation de l'auteur et offrir la visualisation d'une solution cohérente du scénario (Figure 2.9).

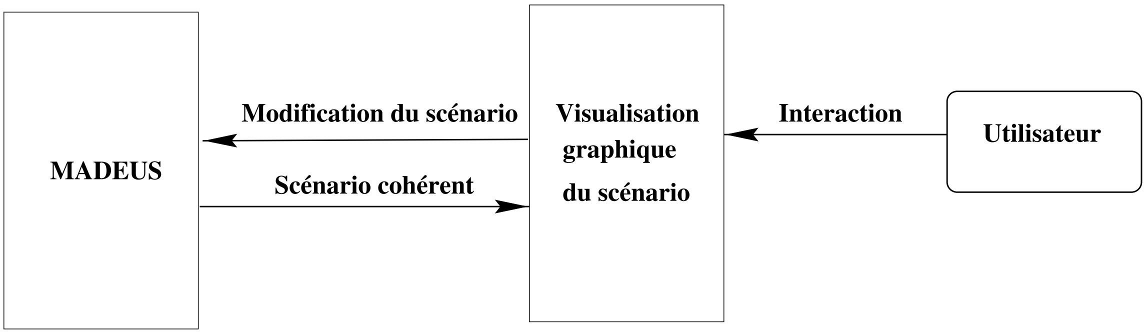


Figure 2.9 : Architecture

Dans le chapitre suivant nous allons présenter une étude sur les différentes approches existantes dans un système de contraintes ou pour le maintien de solution en analysant comment celles–ci peuvent répondre aux différents besoins mis en évidence dans ce chapitre, en particulier pour les problèmes liés au traitement des objets flexibles car nous nous verrons que c'est eux qui posent, du point de vue algorithmique, le plus de problèmes. Nous présenterons dans le chapitre IV une solution qui nous permettra de répondre partiellement à ces besoins et nous permettant ainsi de focaliser notre attention sur les problèmes liés à la représentation graphique du scénario.

Chapitre III

Contraintes

III.1 Introduction

La manipulation des contraintes n’est pas propre à la visualisation graphique du scénario temporel. Que ce soit pour la planification du scénario joué par l’interface de présentation de Madeus [18] ou que ce soit pour le positionnement spatial des objets [5] nous avons besoin de manipuler et de résoudre des systèmes de contraintes. Néanmoins notre problématique soulève des besoins particuliers notamment pour répondre au besoins mis en évidence dans le chapitre précédent.

Au cours de ce chapitre, après un bref rappel de la définition d’un système de contrainte (section III.2), nous présenterons les besoins en terme de résolveurs de contraintes (section III.3) définit à partir des besoins identifiés dans le chapitre II, avant de présenter les différentes approches pour résoudre de tels systèmes (section III.4).

III.2 Résolution de contraintes

Définition d’un système de contraintes

Un problème de satisfaction de contraintes est la donnée d’un ensemble de variables dont les valeurs sont restreintes par des contraintes. Nous allons nous intéresser exclusivement aux CSPs binaires, c’est–à–dire ne manipulant que des contraintes entre deux variables, et au contexte classique dans lequel les contraintes sont discrètes. Le CSP à considérer est un triplé G={X,D,C}, où

– $X = \{x_1, ..., x_n\}$ est un ensemble de variables,

– $D = D_1 \times ... \times D_n$ représente les domaines de ces variables (ensembles de valeurs pouvant leur être affectées *a priori*),

– $C = \{c_{ij} / 1 \leq i,j \leq n\}$ est un ensemble de contraintes d’entrée.

Une contrainte c_{ij} exprimera la contrainte entre les variables x_i et x_j, restreignant du même coup les valeurs pouvant être prises simultanément par ces deux variables. Dans le cas fini, elle s'exprimera généralement sous la forme d'un ensemble de couples de valeurs permises.

$c_{ij} = \{ (x_{i1},x_{j1}), ..., (x_{ip},x_{jp}) \} \subseteq D_i \times D_j$

Dans le cas des domaines infinis la contrainte est donnée en intention par un prédicat sur les variables de la contrainte.

On définit ensuite une solution d'un CSP comme étant une instanciation de toutes les variables satisfaisant l'ensemble des contraintes, c'est–à–dire:

Une solution est un ensemble $\delta = \{x_1 \in D_1, ..., x_n \in D_n \ / \ \forall c_{ij} \in C, (x_i,x_j) \in c_{ij}\}$

Expression du scénario temporel sous forme de contraintes

Dans le cas d'un scénario temporel exprimé sous forme de contraintes, chaque objet est défini par deux variables (ses instants de début et de fin), le domaine de ses variables est $[0, +\infty]$. Chaque objet flexibles introduit une contrainte correspondant à son intervalle de durées possibles :

- BorneInf ≤ Fin – Début ≤ BorneSup

Et chaque relation du scénario est traduite en terme de contraintes qui sont toutes sous la forme d'une différence de la forme :

$$\min \leq X - X' \leq \max$$

min et max pouvant prendre leur valeurs dans l'intervalle $[0.. \infty [$, et où X et X' sont des instants de début ou de fin.

Par exemple la relation During entre les objets 1 et 2 sera traduite par :

- $1 \leq \text{Fin2} - \text{Fin1} \leq +\infty$
- $1 \leq \text{Début1} - \text{Début2} \leq +\infty$

Satisfaction de contraintes ou maintien de solution

Etant donné un système de contraintes, on peut identifier deux problèmes différents :

- **Satisfaction de contraintes**

  On parle de satisfaction de contraintes lorsque l'on cherche à assurer la cohérence d'un système de contraintes et à trouver une ou toutes les combinaisons de valeurs des variables qui permettent de satisfaire l'ensemble des contraintes.

- **Maintien de solution**

  On parle de maintien de solution lorsqu'à partir d'une solution d'un système de contraintes et d'une perturbation de cette solution (soit par modification de la valuation d'une des variables, soit par ajout ou retrait d'une contrainte) on cherche à calculer une nouvelle solution. Dans ce cas, le système de contraintes est dit dynamique.

Notre problématique de visualisation du scénario temporel telle que nous l'avons présenté dans les deux premiers chapitres se situe clairement dans une problématique de maintien de solution d'un système de contraintes dynamiques. En effet à un instant donné on visualise une solution d'un ensemble de contraintes et on doit réagir à une perturbation qui dans le cas présent est la modification de la durée d'un objet. Les besoins utilisateur généralement mis en avant lors de l'étude de cette classe de problèmes est l'efficacité et la stabilité des solutions (ie, la nécessité de modifier le moins possible la solution courante). Nous allons voir dans la section suivante comment ils correspondent aussi à une réalité dans notre contexte mais aussi comment les besoins identifiés en appellent d'autres.

III.3 Les besoins

Nous allons voir dans cette partie comment les besoins identifiés dans le chapitre II se traduisent sur les algorithmes de maintien de solutions.

- **Efficacité du calcul de la réponse**

  Une qualité importante d'une interface est sa capacité à répondre rapidement aux actions de l'utilisateur. Une non–réponse du système dans un délai raisonnable peut amener l'utilisateur à se lasser et à ne plus utiliser le système ou il peut l'amener à répéter sa commande, ce qui peut avoir des effets indésirables. L'efficacité de la méthode de résolution est donc un facteur très important dans notre contexte. Cette efficacité se traduit généralement par l'utilisation de techniques incrémentales (ie, elles profitent de la solution courante pour calculer la nouvelle solution). Là où un résolveur non–incrémental reconsidérerait l'ensemble des contraintes, même en cas de moindre changement, un résolveur incrémental ne réévalue que les contraintes affectées par cette perturbation.

- **Stabilité des solutions, minimisation du nombre d'objets modifiés**

  Nous avons soulevé dans le chapitre II le problème de contrôle de la solution en cas de réponses multiples, ce besoins ce traduit par les deux points suivants :

  - **Mesure de distance entre les variables**

    Cela nécessite que le résolveur de contrainte permette l'utilisation d'une fonction de localité entre les variables. Par exemple on peut imaginer de définir une fonction mesurant la distance entre deux variables en tenant compte de l'existence d'une relation directe entre deux objets.

  - **Priorité sur les variables à déformer**

    Il est souhaitable que le système de contrainte puisse nous permettre de définir des priorités sur les variables à modifier, ou sur contraintes, ceci afin de pouvoir privilégier entre autre la déformation des délais par rapport aux autres objets.

- **Connaissance de l'intervalle de validité d'une variable**

  L'intervalle de validité d'une variable est l'ensemble des valeurs permises pour cette variable telles qu'il existe une valuation pour les autres variables qui rendent le scénario cohérent. Cette information nous permettrait d'anticiper la flexibilité des durées des objets.

III.4 Les solutions existantes

On peut diviser les méthodes de maintien de solution en deux catégories : les approches locales et les approches globales.

III.4.1 Les approches locales

Les résolveurs locaux utilisent un graphe de contraintes tel celui présenté dans la Figure 3.1 :

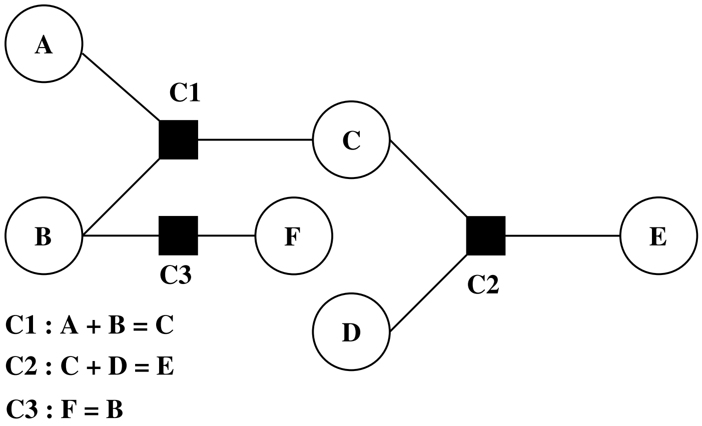


Figure 3.1 : Graphe de contraintes

Le graphe de contrainte est construit de la manière suivante. Chaque variable du problème est représentée par un cercle, chaque contrainte est représentée par un carré, et chaque arc représente l'appartenance d'une variable a une contrainte. A chaque contrainte est associée un ensemble de méthodes qui explicite comment répercuter la modification d'une variable sur les autres variables présentes dans la contrainte. Par exemple, une méthode associée à la contrainte C1 peut être $A \leftarrow C + B$ qui indique comment répercuter une modification de C et/ou B (ce sont les entrées de la méthode) sur A (c'est la sortie de la méthode).

C'est la technique de résolution la plus employée dans les applications graphiques. De nombreux travaux ont été fait sur le sujet et se poursuivent (ex : ThingLab [27] et SketchPad [2]).

La fonction d'un résolveur local incrémental est de décider :

- les contraintes à réévaluer, seules celles qui risquent d'être violées par la perturbation et par ses conséquences sont reconsidérées, c'est en cela que ces techniques sont incrémentales.
- les méthodes à utiliser, pour chaque contraintes à réévaluer, il faut décider quelle est la méthode associée qui va permettre de la satisfaire à nouveau (en fonction des choix déjà effectués sur les autres contraintes).

- L’ordre dans lequel celles–ci doivent être appliquées afin de satisfaire à nouveau l’ensemble des contraintes.

La méthode la plus communément utilisée pour assurer ces trois fonctionalités est la méthode par propagation locale de valeurs [25]. L’idée sous–jacente à cette technique est de dire que dès que la contrainte possède assez d’informations pour calculer des valeurs, elle les calcule. Elle se décompose en deux phases :

1. Une phase de planification au cours de laquelle le résolveur sélectionne une méthode pour chacune des contraintes à recalculer, et uniquement pour celles–ci. Cet ensemble de méthodes est calculé en partant de la variable perturbée et en identifiant toutes les contraintes potentiellement insatisfaite. Pour chacune de ces contraintes une méthode est choisie. Cette méthode doit prendre la variable perturbée en entrée et une variable ne doit pas être en sortie de deux méthodes. Ce choix construit un ensemble de variables modifiées par la perturbation initiale. Il faut donc calculer un nouvel ensemble de contraintes à réévaluer parmi les contraintes non encore examinées, et recommencer ainsi de suite jusqu’à ne plus avoir de contraintes à reconsidérer. L’ordre d’exécution correspond a un tri topologique de ce graphe en fonction de son orientation.

   Par exemple une orientation possible dans l’exemple de la Figure 3.1 est donnée dans la Figure 3.2, cette orientation répercute une modification de A sur les variables C, D et F.

2. Une phase d’exécution pendant laquelle le plan constitué est exécuté en séquence . Dans cette phase, les valeurs connues sont propagées le long des arcs du graphe de contraintes.

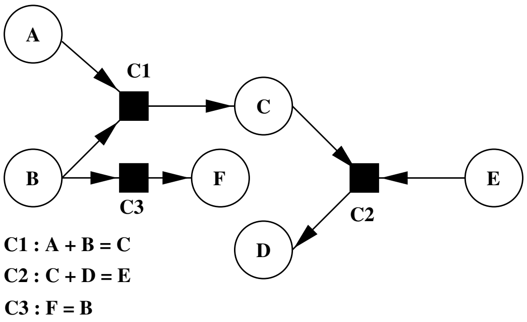


Figure 3.2 : Orientation du graphe de contraintes

Les avantages de la propagation locale sont l'efficacité, la généricité et les facilité de compréhension du comportement des variables. Les désavantages bien connus de ces méthodes sont :

- L'incomplétude de la méthode lorsque par exemple le graphe de contrainte est cyclique. Par conséquent de tels systèmes rejettent les systèmes de contraintes qui introduisent des cycles. Dans notre contexte, les systèmes de contraintes que l'on manipule engendrent des graphes de contraintes fortement cycliques.
- La manipulation de contraintes uniquement fonctionnelles. Une contrainte à n variables est fonctionnelle, si lorsque l'on fixe n−1 variables, la dernière est déterminée de façon unique. Dans notre contexte, les contraintes qui limitent l'intervalle de validité d'une variable ne sont pas fonctionnelles.

Ces deux raisons sont suffisantes pour ne pas retenir ces méthodes dans notre contexte. Cependant les techniques employées, comme la définition de poids associés aux contraintes, pour contrôler la solution et orienter la recherche sont intéressantes et adaptable à notre problème. Pour ce qui est de notre besoin de connaître les domaines de validité des variables, nous n'avons à ce jour pas d'idée sur la façon d'obtenir ces informations en utilisant ces méthodes sauf par des essais successifs.

III.4.2 les approches globales

Nous venons de voir que les méthodes locales ne permettent pas de répondre complètement à nos besoins spécifiques. Nous étudions à présent l'autre type d'approche, les approches globales. On ne présentera qu'un seul algorithme de cette classe qui est représentatif de l'approche. Cet algorithme appelé Dynamic Backtrack est une adaptation de l'algorithme backtrack utilisé dans un problème de satisfaction statique de contraintes. Celui ci est présenté ci dessous.

L'algorithme de backtrack

Nous rappelons simplement les caractéristiques essentielles de cet algorithme. Dans l'état initial aucune variable n'est affectée. A une étape donnée l'ensemble des variables sera divisé en deux groupes, les variables affectées et les variables non affectées. La solution courante est cohérente avec l'ensemble de contraintes. L'étape va consister à affecter une nouvelle variable, le système choisi une variable et une valeur dans l'ensemble de celles possibles et vérifie la cohérence. Si le système est cohérent il passe à l'étape suivante, si le système est incohérent il essaie une autre valeur pour la variable considérée. Si toutes les valeurs ont été considérées il désaffecte une des variables affectées, et repart à l'étape n−1.

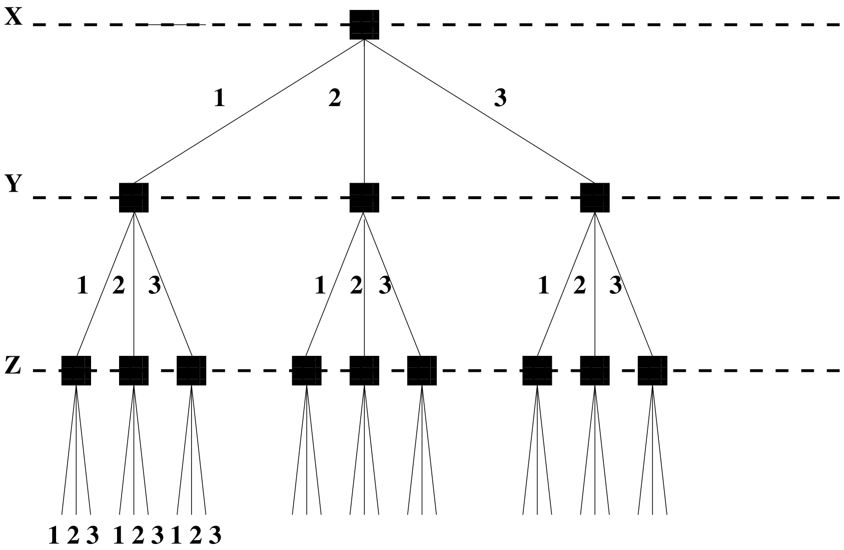


Figure 3.3 : Arbre de recherche

Dans l'exemple de la Figure 3.3 on peut voir un arbre de recherche possible dans le cas où nous avons des contraintes qui portent sur trois variables X,Y et Z (chacune étant définie dans l'intervalle [1..3]).

C'est par exemple la méthode employée dans IlogSolver [15].

Retour arrière dynamique

Cet algorithme part d'une solution courante, son mécanisme est assez proche de la version de l'algorithme de backtrack présenter précédemment. Cet algorithme divise les variables en deux catégories: variables fixées et variables non fixées. Dans l'état initial aucune variable n'est fixée. A chaque étape toutes les variables sont valuées, et le système peut être ou non cohérent, mais la valuation des variables fixées est cohérente. A chaque étape le système cherchera à étendre l'ensemble des variables fixées. Des heuristiques sur le choix des variables à remettre en cause lors de retour arrière ou sur le choix des valeurs pour une variable sont réalisable, ce qui permet d'améliorer et de personnaliser cet algorithme de manière à orienter la recherche de solution. Par exemple on peut imaginer, dans notre cas, de désaffecter en priorité les variables représentant les instants de début et fin des délais.

L'avantage majeur de cette technique est sa complétude: on trouve toujours la solution si elle existe et la possibilité pour l'utilisateur de contrôler la recherche de solution en cas de réponse multiples.

Ses inconvénients sont :

- l'obligation de travailler sur des domaines finis, or nos systèmes de contraintes manipulent des domaines infinis.
- le coût généralement très grand de cette technique, qui procède par énumération, lorsque le domaine des variables est trop grand.

Cependant ces deux inconvénients peuvent être réduits par les deux remarques suivantes :

- Il est possible d'utiliser sur nos graphes temporels un algorithme de filtrage appelé PC2 [21] qui va réduire les domaines de validité de toutes les variables aux valeurs qui apparaissent dans au moins une solution du système de contraintes, et donc de diminuer l'espace de recherche améliorant ainsi les coûts d'un algorithme de type backtrack dynamic. Un autre avantage de l'utilisation d'un algorithme de type PC2, et qu'il permet de connaître l'intervalle de validité des variables.
- Nous posons la conjecture suivante : toute modification élémentaire (+1 ou −1 unité) d'une variable se traduit par au plus une modification élémentaire sur les autres variables. Cette conjecture nous permet de limiter tous les domaines à des déformations élémentaires autour de la solution courante, éliminant ainsi le problème des domaines infinis. Cette conjecture bien entendu reste à démontrer.

III.5 Conclusion

Comme nous avons pu le voir au cours de ce chapitre aucune méthode de maintien de solution répond complètement à nos besoins :

- les méthodes locales du fait de leur incomplétude et par leur incapacité à manipuler des contraintes non fonctionnelles ;
- les méthodes globales du fait de leur manque d'efficacité.

Néanmoins dans le cas de approches globales nous avons quelques idées pour contourner ces difficultés, notamment avec l'application d'algorithmes de type PC2.

Un problème commun à toute ces approches (locale et globale) est quelles ne nous permettent pas d'anticiper l'intervalle de flexibilité. Il va dont être nécessaire de réaliser un algorithme qui nous permettra d'anticiper cette intervalle dans toutes les configurations possibles (déplacements locaux ou globaux) et de manière efficace.

Nous allons maintenant présenter une méthode ad'hoc qui nous permet de traiter de manière efficace une partie des besoins présentés ainsi que les choix de représentation graphique pour les concepts de flexibilité et d'interruption présentés dans le chapitre II.

Chapitre IV

Mise en œuvre

IV.1 Une solution ad’hoc pour le maintien de solution

IV.1.1 Pourquoi une solution ad’hoc

Comme nous l’avons dit dans le chapitre II, nous avions besoin de maintenir localement la solution courante de notre ensemble de contraintes définissant le scénario temporel d’un document multimédia. De plus nous avons besoin de contrôler le système de résolution de manière à pouvoir définir des préférences sur les méthodes à utiliser ainsi que sur les objets à déformer. Dans l’attente de trouver un résolveur permettant de résoudre de façon plus générale (possibilité d’utiliser l’algorithme à la fois pour le placement spatial, placement temporel, et pour la résolution du système de contrainte), nous avons adapté et développé un algorithme réalisé dans le cadre du DEA et fondé sur l’utilisation du graphe temporel (qui peut être considéré comme le graphe de solution correspondant à l’ensemble des contraintes) plutôt que sur le graphe de contraintes de manière à utiliser pleinement la topologie du problème ainsi que de sa solution. Cet algorithme développé dans le cadre du DEA ne prenait en compte que les perturbations liées au déplacement d’un objet dans le représentation graphique du scénario temporel. Nous l’avons retravaillé afin de traiter le cas d’une perturbation par retaillage.

Dans un premier temps nous allons présenter le graphe temporel sur lequel repose nos algorithmes, dans un second temps nous présenterons nos algorithmes avant de présenter dans un troisième temps leur intégration et leur utilisation dans le système de visualisation graphique du scénario temporel.

IV.1.2 Le graphe temporel

Le graphe temporel est un graphe acyclique orienté. Les nœuds sont les instants de début et de fin des intervalles. La présence d'un arc entre deux nœuds n1 et n2, traduit le fait que n2 se situe temporellement après n1. Les arcs sont étiquetés par 4 valeurs:

- la durée prévue de l'objet
- la durée effective de l'objet, en effet à cause des opérateurs d'interruption la durée effective peut être différente de la durée prévue.
- la borne inférieur de l'intervalle de flexibilité de l'objet dans le scénario
- la borne supérieure de l'intervalle de flexibilité de l'objet dans le scénario

Pour expliquer la construction du graphe temporel (pour plus de détails voir [19]), nous allons donner l'idée de sa construction incrémentale.

Nous distinguons deux types d'actions :

- Insertion et création d'un objet (cadre 1 de la Figure 4.1) :

  L'objet est représenté pas un intervalle. Nous créerons donc un arc entre deux nœuds.

- Insertion d'une relation:

  L'insertion d'une relation entre deux objets a pour effet de modifier le graphe temporel. Ces modifications peuvent être de deux types:

  - Fusion de deux instants (cadre 2 de la Figure 4.1), c'est par exemple le cas pour les relations start, end, equal...
  - Insertion d'intervalle (cadre 3 de la Figure 4.1), c'est par exemple le cas pour les relations before, after, during ...

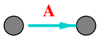
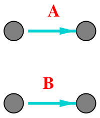
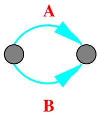
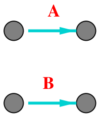
ETAT INITIAL	ETAT FINAL	ACTIONS
1		CREATION d'un OBJET
2 		INSERTION d'une RELATION A EQUAL B
3 		INSERTION d'une RELATION A BEFORE B

Figure 4.1 : Construction du graphe temporel

Dans la Figure 4.2 nous présentons un exemple de graphe temporel.

Représentation Symbolique

Représentation Interne

Doc {

A Starts B

B Equal C

B Before D

D Starts F

D Before(δ3) E

F During A

F Finish E

}

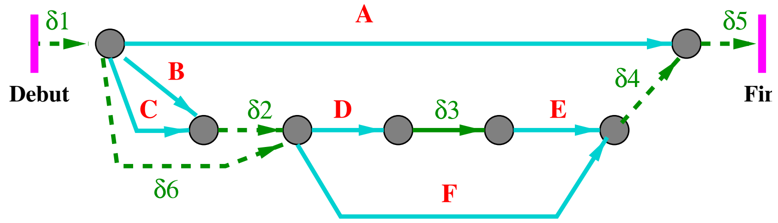


Figure 4.2 : Le graphe temporel interne

IV.1.3 Algorithmes de maintien de solution

L'objectif est de trouver à partir d'une solution donnée et d'une perturbation de l'auteur (retaillage ou déplacement) une nouvelle solution cohérente. De plus on veut pouvoir empêcher a priori toutes perturbations qui conduisent à un système incohérent.

Le premier algorithme que nous allons présenter est une optimisation de l'algorithme réalisé dans le cadre de mon DEA. Cet algorithme suppose l'existence d'une solution courante au système de contraintes, cette solution nous est donnée par exemple par l'utilisation d'un algorithme de type PathConsistency [21].

Que ce soit pour le déplacement ou le retaillage l'idée est la même : lorsque l'objet est sélectionné il faut calculer l'intervalle de déplacement ou de retaillage permis (c'est de cette manière que l'on sait a priori quelles sont les perturbations autorisées) ainsi que la liste des objets qui vont être modifiés (déplacement, retaillage) par la perturbation.

Il est important de noter que ces calculs ne se font qu'au moment de la sélection de l'objet, au cours du déplacement on ne fait que mettre à jour les informations graphiques pour l'affichage des objets. Ce principe nous permet d'obtenir de bonnes performances en temps.

IV.1.3.1 Déplacement d'un objet

L'idée générale de l'algorithme est de parcourir tous les chemins entrants ou sortants des deux nœuds de l'intervalle représentant l'objet sélectionné et de trouver, pour chacun de ces chemins, un objet amortissant les déplacement appliqués par l'auteur sur cet objet. Dans cette version, nous recherchons le premier délai flexible sur chaque chemin.

L'algorithme proposé se décompose en deux phases:

- Identification des objets se déplaçant avec l'objet sélectionné et des objets amortissant le déplacement de l'objet sélectionné.
- Classement des objets amortissants en fonction de leur réactions (élongation ou rétraction) aux déplacements vers la gauche ou vers la droite de l'objet sélectionné. L'objectif est de regrouper les objets amortissant qui vont avoir la même réaction (élongation ou rétraction) lors d'un déplacement vers la droite ou vers la gauche de l'objet sélectionné. Le résultat est un couple de listes l'une (respectivement l'autre) contient les objets amortissants qui vont s'allonger lors d'un déplacement à droite (respectivement à gauche) de l'objet sélectionné et se rétrécir lors d'un déplacement à gauche (respectivement à droite) de l'objet sélectionné.

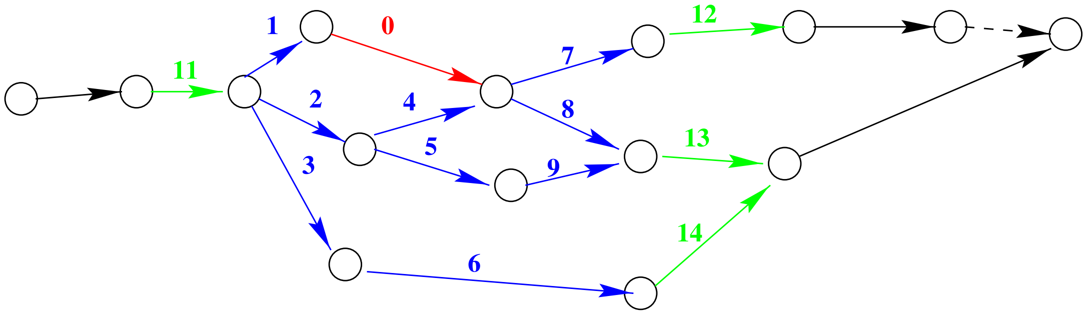


Figure 4.3 : Parcours du graphe temporel

Par exemple, dans la Figure 4.3 nous représentons un exemple de parcours possible, l'objet sélectionné est en rouge, les objets qui vont se déplacer sont en bleu et les objets qui vont amortir le déplacement sont en vert. Dans ce cas les listes obtenues sont les suivantes:

- Liste des objets qui bougent : 0 1 2 3 4 5 6 7 8 9 10
- Liste des objets qui amortissent le déplacement : 11 12 13 14

Dans la seconde phase la deuxième liste sera divisée en deux :

- L1 : 11
- L2 : 12, 13, 14

L1 (respectivement L2) est la liste des objets qui vont s'allonger (respectivement se rétrécir) lors d'un déplacement à droite (respectivement à gauche) de l'objet sélectionné et se rétrécir (respectivement s'allonger) lors d'un déplacement à gauche (respectivement à droite).

Avant de présenter les algorithmes nous allons présenter rapidement certaines fonctions qui nous seront utilisées par la suite.

Fonction : ADroite

```
        Variables
              Objet 1 (Nœud11,Noeud12)
              Objet 2 (Nœud21, Nœud22)
        Renvoie vrai si L'objet 2 est à droite de l'objet 1. C'est a
        dire qu'il appartient soit a la liste des arcs entrant dans
        Nœud21 ou Nœud22.
```

Fonction : ∈ TypeAmorti

Variable

`Objet1`

`Renvoie vrai si L'objet 1 est considéré comme un objet amortissant.`

`Dans notre implémentation ce sont les objets de type délais flexibles, mais il est envisageable d'utiliser d'autres définitions pour généraliser l'algorithme.`

Fonction : enRelationAvec

Variable

`Objet1`

`Renvoie l'ensemble des objets qui sont en relation directe (il existe une contrainte directe) avec l'objet 1 (ie : il existe une contrainte qui lie les instants de début et/ou de fin des deux objets)`

Fonction : enRelationAvecLaFinDe

Variable

`Objet1`

`Renvoie l'ensemble des objets qui sont en relation directe (il existe une contrainte directe) avec la fin de l'objet 1 (ie : il existe une contrainte qui lie les instants de fin des deux objets)`

Nous allons maintenant présenter l'algorithme qui identifie les objets affectés par le déplacement de l'objet sélectionné:

Fonction : IdentificationDesObjetsAffectés

```
Variables
    Entrées :
        ObjetSel
```

```
                  LObjetsAmortis
                  LObjetsDeplacés
        Locales :
                  Objet
        Sorties :
                  LObjetsAmortis
                  LObjetsDeplacés

Début
        PourChaque Objet enRelationAvec ObjetSel
            Si $\neg$ EstMarqué(Objet) alors
              Marquer(Objet)
              Si Type(Objet) $\in$ TypeAmorti alors
                      Objet $\rightarrow$ LObjetsAmortis
              Sinon
                  IdentificationDesObjetAffectés
                  (Objet,LObjetsAmortis,LObjetsDéplacés)
                  Objet $\rightarrow$ LObjetDeplace
              Fin Si
            Fin Si
        Fin Pourchaque
Fin
```

Nous allons maintenant présenter la fonction qui permet de classer les objets affectés :

Fonction : Classification :

```
    Variables
        Entrées :
                  ObjetS
                  LObjetsAmortis
                  LObjetsDeplacés
        Locales :
                  Objet
        Sorties :
                  LAmortiD
                  LAmortiG

    Début
        PourChaque Objet de LObjetsAmortis
            Si $\exists$ O telque
              (O $\in$ LObjetsDeplacés et
               Adroite(O, Objet)) alors
```

```
                Objet → LAmortisD
            Sinon
                Objet → LAmortisG
            Fin Si
        Fin Pourchaque
    Fin
```

IV.1.3.2 Redimensionement

L'algorithme pour le redimensionement des objets est fortement inspiré de cet algorithme, seul la première passe est différente. Lors du calcul des objets affectés il faut gérer une détection des cycles dans le graphe temporel. De plus le premier appel est différent des autres, en effet on ne lance la récursivité que sur un des deux nœud de l'objet sélectionné en fonction de la manipulation effectuée.

Nous allons maintenant présenté le premier appel pour la recherche des objets affectés par la manipulation, la récursivité se fera sur une fonction très similaire à la première étape présentée précédemment avec en plus le test pour la détection des cycles.

Fonction : DetectionCycle

```
         Variable
                Objet
         Renvoie vrai si on est sur un cycle et tous les autres objets
         appartenant a ce cycle sont marqué.
```

Fonction : Premiere_Etape_Retaillage

```
    Variables
        Entrées :
                ObjetS
                LObjetsAmortis
                LObjetsDeplacés
        Locales :
                Objet
        Sorties :
                LObjetsAmortis
                LObjetsDeplacés
```

```
Début
   Si RetaillageBordDroit alors
     L ←  EnRelationAvecLaFinDe ObjetS
   sinon
     L ← EnRelationAvecLeDebutDe ObjetS
   FinSi
     PourChaque Objet de L
         Si ¬EstMarqué(Objet) alors
           Marquer(Objet)
            Si (Type(Objet) ∈ TypeAmorti
               ou DetectionCycle) alors
                     Objet → LObjetsAmortis
             Sinon
                IdentificationDesObjetsAffectés (Objet,
                LObjetAmortis, LObjetsDéplacés)
                Objet → LObjetDeplacés
            Fin Si
           Fin Si
     Fin Pourchaque
Fin
```

Dans les deux cas le coût de l'algorithme est de l'ordre de O(n) avec n le nombre d'objets (objets + délais) du graphe au niveau de hiérarchie courant, ce qui est très satisfaisant.

IV.1.4 apports et limites de la solution

Cette solution répond à trois des critères que nous avons identifiés :

- **Efficacité du calcul de la réponse :**

  les coûts sont en O(n) ce qui nous permet d'assurer un déplacement continu aux mouvements de la souris près.

- **Définition des objets à déformer :**

  L'utilisation de la fonction ∈ TypeAmorti permet de définir les objets que l'on souhaite déformer. Cette définition peut être faite au niveau du type des objets ou au niveau des objets même.

- **Anticipation de l'intervalle de validité d'une variable :**
  la fonction définie permet de calculer cet intervalle en fonction des objets déformables, à condition que nous limitons les déformations aux premiers objets déformables rencontrés.

Néanmoins cet algorithme suppose que les déformations se limitent au niveau de hiérarchie courant. Ce choix a été fait dans le souci de limiter les perturbations visuelles en permettant ainsi à l'auteur de prévoir et de comprendre toutes les modifications effectuées.

Cette solution ne permet pas de :

- répercuter les déformations sur plusieurs objets. Par exemple, sur la Figure 4.4 si l'auteur agrandit l'objet A par la droite, les déformations ne peuvent être faites simultanément sur B et C.

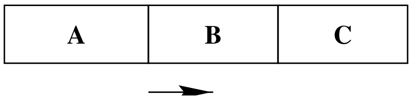


Figure 4.4 : Impossibilité de répercuter simultanément les déformations sur 2 objets

- répercuter les déformations sur plusieurs niveaux de hiérarchie. Par exemple, sur la Figure 4.5 si l'auteur agrandit l'objet A par la droite, les déformations ne peuvent être faites sur l'objet Scène 2, les déformations ne seront permises que sur les objets présents au niveau Scène 1.

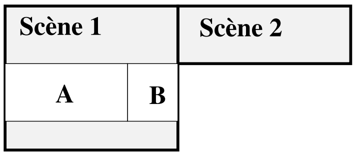


Figure 4.5 : Impossibilité de répercuter simultanément les déformations sur 2 niveaux de hiérarchie différents

Dans les deux cas, pour résoudre ces problèmes, on peut imaginer un mécanisme qui permettrait d'appeler plusieurs fois de suite l'algorithme actuel, et qui puisse connaître les objets indéformables pour une manipulation donnée. Cependant connaître cette information a priori n'est pas triviale et risquerait d'entraîner une importante dégradation des performances.

IV.2 Flexibilité

Nous venons de voir que nous avons maintenant à notre disposition un algorithme qui nous permet de manipuler la flexibilité des objets. Nous allons donc nous intéresser maintenant à l'intégration de cette flexibilité dans le système de visualisation.

IV.2.1 Choix de représentation : deux métaphores complémentaires

Dans le souci de ne pas surcharger la représentation graphique, et de respecter l'honnêteté temporelle [28] nous avons choisi d'utiliser deux métaphores graphiques pour représenter cette flexibilité.

La première qui consiste à représenter explicitement les bords retaillables des objets, cette représentation est toujours présente à l'écran. La seconde consiste à changer le curseur de la souris pour montrer à l'auteur qu'il peut retailler l'objet. Cette métaphore, contrairement à la première n'est donc présente que lorsque le curseur souris est sur un des bords retaillable de l'objet.

IV.2.2 Réalisation

Dans l'exemple de la Figure 4.6 nous pouvons voir l'affichage avant et après manipulation de la flexibilité de l'objet Texte 3. L'auteur a agrandi de 1 cet objet par la droite. Le scénario s'est adapté à cette modification de façon continue, et nous pouvons voir que tous les objets flexibles sont facilement identifiables.

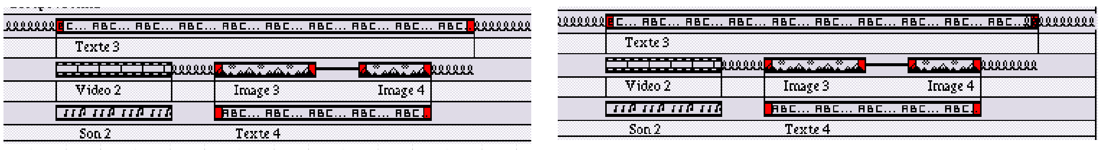


Figure 4.6 : Manipulation de la flexibilité des objets

L'algorithme décrit dans la section précédente a identifie l'objet à déformer et a répercuter les déformations nécessaires pour compenser l'agrandissement de l'objet Texte3.

IV.3 Intégration des opérateurs d'interruption

IV.3.1 Visualisation des opérateurs d'interruption

Comme pour la visualisation des relations d'Allen (section I.2.1) nous avons choisi de représenter explicitement les opérateurs d'interruption (comme nous l'avions suggéré dans le chapitre II).

Nous représentons aussi explicitement les objets coupés (par une représentation en grisé) de manière à ce que l'auteur puisse se rendre compte que des objets n'ont potentiellement pas totalement délivrés leur contenu sémantique.

IV.3.2 Réalisation

Sur l'exemple de la Figure 4.9 nous pouvons voir la représentation de l'opérateur Parmin pour les objets définis dans la Figure 4.7 et les relations définies dans la Figure 4.8. Nous pouvons aussi voir explicitement la partie des objets interrompue.

Objets	**Durées**	
	Effectives	**Prévues**
Objet 1	5	5
Objet 2	5	6
Objet 3	6	6
Objet 8	7	7
Objet 10	5	6
Délai Before	3	3

Figure 4.7 : Relations entre les objets et durées des objets

Relations

Objet 1 `Start` Objet 8

Objet 1 `Parmin` Objet 2

Objet 1 `Start` Objet 10

Objet 1 `Before` Objet 3

Objet 2 `Equal` Objet 10

Figure 4.8 : Scénario

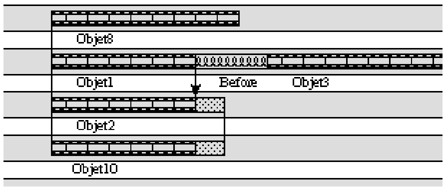


Figure 4.9 : Visualisation des opérateurs d'interruption

IV.4 Conclusion

La représentation graphique de la flexibilité permet à l'auteur de savoir quels sont les objets déformables, mais aussi quelles sont les manipulations qu'il peut effectuer sur le scénario courant.

La représentation des opérateurs d'interuption permet elle d'anticiper le comportement du scénario et de comprendre la solution courante, tout en ayant la connaissance des effets de l'interruption, notamment sur les objets interrompus.

La solution algorithmique proposée pour la déformation de solution a permis de tester les idées graphiques, mais il faut chercher une solution plus générale, notamment pour permettre des déformations globales et intégrer des déformations sur plusieurs niveaux de hiérarchies.

Conclusion

1 Synthèse

La spécification du scénario temporel d'un document multimédia occupe une place majeure lors de l'édition d'un tel document, de part son caractère novateur (cette notion de scénario n'existe pas dans les documents usuels) et sa complexité.

Différentes approches ont été proposées pour mener à bien cette tâche qui vont de l'utilisation d'un langage de script à l'utilisation d'un langage déclaratif à base de contraintes, en passant par l'utilisation de structures graphiques comme les arbres ou les réseaux de Pétri. Au cours de ce stage, notre intérêt s'est porté sur les approches fondées sur l'utilisation de contraintes et plus précisément sur celle proposée par Madeus l'environnement d'édition/présentation de documents multimédia développé au sein du projet Opéra.

Remarquons que le travail effectué au cours de ce stage bien qu'il ait été fait dans le cadre de l'environnement Madeus, s'adapterait aisément aux autres environnements d'édition fondés sur les contraintes d'intervalles : Isis [45] et HPAS [30].

Dans ce contexte, un premier travail a été réalisé dans le cadre du DEA visant à définir les bases d'une vue graphique du scénario temporel qui soit une véritable aide pour l'auteur. Nous avons souhaité aller plus loin que la définition d'un équivalent graphique à chaque contrainte, comme cela est proposé dans les autres environnements d'édition de documents multimédia à base de contraintes. Cette représentation graphique se veut être utile et efficace en offrant une réponse visuelle aux questions que se pose l'auteur sur le plan de la compréhension et de l'analyse du scénario temporel du document multimédia.

Dans le cadre de ce magistère nous avons étendu cette représentation de manière à ce quelle intègre la flexibilité des objets et les opérateurs d'interruption. Ces deux aspects sont particulièrement importants pour la conception d'un scénario puisqu'ils introduisent respectivement :

- une grande capacité d'adaptation du système d'édition aux spécifications de l'auteur ;

- la possibilité de définir des règles d’interruption d’un média sur un autre.

Ces deux aspects avaient, dans un premier temps, été mis de coté afin de réduire la complexité du problème à traiter. Pour effectuer cette intégration nous avons été amenés à identifier les besoins liés à l’introduction de ces deux aspects dans le système de visualisation et d’autre part à analyser comment le domaine de la résolution de systèmes de contraintes pouvait offrir une réponse satisfaisante à ces besoins. Il faut savoir que ce domaine d’étude recouvre des problèmes très différents (contraintes sur des domaines finis, contraintes sur des domaines infinis, satisfaction et maintien de cohérence, techniques de filtrage, contraintes temporelles, ...). Ce domaine est aussi très riche en algorithmes et fait toujours l’objet de nombreuses recherches. Par conséquent, une des réelles difficultés du stage a été de se frayer un chemin dans ce labyrinthe.

Nous avons à présent les connaissances pour mieux positionner notre problématique et les problèmes qu’elle soulève face aux travaux de la communauté "résolution de contraintes".

2 Réalisation

En parallèle avec cette analyse des besoins et des solutions potentielles nous avons défini et implémenté deux algorithmes (un algorithme de maintien de solution, un algorithme qui permet d’anticiper les déformations locales) qui doivent être considérés comme des solutions ad’hoc à un problème particulier de maintien de solution dans un système de contraintes. Ces algorithmes et leurs limites ont été analysées dans le chapitre IV.

La réalisation de ces deux algorithmes nous a permis d’intégrer dans notre prototype plusieurs nouveaux concepts, notamment :

- la visualisation de la flexibilité des objets ;
- la visualisation des opérateurs d’interruption et de leur effets ;
- l’anticipation des déformations permises.

3 Perspectives

Le travail effectué au cours de ce stage doit être vu comme une première étape dans le développement d'une interface d'édition d'un document multimédia par manipulation directe. La problématique concernant le maintien de solution d'un système de contraintes va se retrouver tout au long de ce développement. En effet, dès que l'auteur va vouloir enlever ou ajouter une contrainte, apparaîtra le besoin de perturber le moins possible la solution courante en un temps réduit. Le travail réalisé au cours de ce stage pourra donc être largement réutilisé. A plus court terme la suite logique de ce travail est d'approfondir les pistes qu'il a ouvert:

- démontrer la conjecture présentée dans le chapitre III et définir un algorithme de maintien de solution qui s'appuie sur PC2 ;
- comparer cette solution à la solution ad'hoc dont il faut poursuivre le développement.

Bibliographie

[1] ALLEN J.F., ‘‘Maintening Knowledge about Temporal intervals’’, *Communications of the ACM*, Vol. 26 (No 11), pp. 832−843, november 1983.

[2] BORNING A., ‘‘Thing−Lab− A Constraint−Oriented Simulation Laboratory’’, *PhD thesis*, Departement of Computer Science, Standford university, Standford, CA, 1979.

[3] BUCHANAN C. Zellweger P.T., ‘‘Specifying Temporal Behavior in Hypermedia Documents ’’, *Proceeding of the ACM, Conference on Hypertext*, pp. 262−271, December 1992.

[4] BULTERMAN D. and HARDMAN L., ‘‘Multimedia Authoring Tools: State of Art and Research Challenges’’, Springer LNCS(1000), pp. 575−591, 1995.

[5] CARCONE L., *Gestion de contraintes spatiales dans Madeus*, Mémoire d’ingenieur CNAM, (à paraître), Grenoble 1997.

[6] DEL BIMBO A., VICARIO E., ‘‘Visual Programming of Virtual Worlds Animation’’, *IEEE Multimedia*, Vol 3(n 1), pp. 40−49, Spring 1996.

[7] Director, *Manuel d’apprentissage de DIRECTOR*, MACROMEDIA 1996.

[8] ELLIOT G.J., Jones E., ‘‘Authoring semantic hypermedia: a concept mapping approach ’’, *ED−Media et Ed−Telecom96*, Boston, june 96.

[9] FURUTA R., QUINT V. et ANDRE J., ‘‘Interactively Editing Structured Documents’’, *Electronic publishing − Origination*, 1 (1), pp. 19−44, April 1998.

[10] GOSLING J., *Algebric Constraints*, PhD Thesis, Departement of computer Science Carnegie Mellon university, May 1983.

[11] GRAM C.and COKTON G., *Design Principles for interactive software*, Christian Gram and Gilbert Cockton, Chapitre 5 pp 133–187.

[12] HARDMAN L., VAN ROSSUM G., BULTERMAN D.C.A., ‘‘Structured Multimédia Authoring’’, *Proceedings of the first ACM International Conference on Multimédia*, pp. 283–289, Anaheim , California, August 1993.

[13] HODGES M.E., HEINES R.M. and ACKERMAN M.S., ‘‘A Construction Set for Multimedia Applications ’’, *IEEE Software*, 6 (1), pp. 37–43, Janvier 1989.

[14] HyTime Information Technologie, ‘‘Hypermedis/ Time–based Structuring Langage (HyTime)’’, *ISO/IEC 10743*, November 1992.

[15] ILOG SOLVER, ‘‘Reference Manual’’, 1995.

[16] JOURDAN M., LAYAIDA N., SABRY–ISMAIL L., ‘‘Time Representation and Management in MADEUS: an authoring environment for multimedia documents’’, *Multimedia Computing and Networking 1997*, pp. 68–79, February 1997.

[17] KIM M. and SONG J., ‘‘Multimedia Documents with Elastic Time’’, *Proceedings of the third ACM International Conference on Multimedia, edite par ACM Press*, pp. 143–154, Polle Zellweger, San Fransisco, Californie 5–9 novembre 1995.

[18] LAYAIDA N, SABRY–ISMAIL L., ‘‘Maintaining Temporal Consitency of Multimédia Documents using Constaint Networks’’, *Multimedia Computing and Networking 1996*, M. Freeman, P. Jardetzky H. M. Vin ed.pp. 124–135, January 1996.

[19] LAYAIDA N., *Madeus: un système d’édition et de présentation de documents structurés multimédia*, Thèse de doctorat, Université Joseph Fourier, Juin 1997 (à paraître).

[20] LAYAIDA N., VION–DURY J.Y., ‘‘Multimedia Authoring: A 3D Interactive Visualization Interface based on a Structured Document Model’’, *Proceedings of the Sixth International Conference on Human–Computer Interaction*, Elsevier Science Publishers,Yokohama, Japon, Juillet 1995.

[21] MACKWORTH A.K. et FREUDER E.C., ‘‘The Complexity of Some Polynomial Network Consistency Algorithms for constraint Satisfaction Problems’’, *Artificial Intellligence*, 25 (1), pp. 65–74, 1985.

[22] OUSTERHOUT J.K., *Tcl and the Tk Toolkit*, Addison–Wesley, 1993.

[23] QUINT V., Vatton I., ‘‘Combinig Hypertext and Structured Documents in Grif’’, *Proceeding of the ACM, Conference on Hypertext*, pp. 23–32, December 1992.

[24] SABRY L., *Langage de specification de la structure temporelle des documents multimedia*, DEA Informatique, Université Joseph Fourier, Grenoble , Juin 1995.

[25] SANNELA M., ‘‘The skyBlue Constraint Solver’’, *TR. 92–07–02 Dept. of computer science and engeenering University of Washington*, 1992

[26] SONG J., KIM M., RAMALIGAN G. et MILLER R., ‘‘Interactive Authoring of Multimedia Documents’’, *Visual Language 96* , 1996.

[27] SUTHERLAND I.E., ‘‘Sketchpad, a man machine graphical communication system ’’, *procedings of the Spring joint Computer Conference* , pp. 329–346, Detroit Michigan USA 21–23 may 1963.

[28] TARDIF L., *DEA d’Informatique*, Visualisation du scénario temporel d’un document multimédia, Université Joseph Fourier, Juin 1997.

[29] Tcl/Tk , *http://sunscript.sun.com/.*

[30] VAN ROSSUM G. and al, ‘‘CMIFed: A presentation environment for portable hypermedia documents ’’, *Proceedings of the first ACM International Conference on Multimédia*, pp. 183–188, august 1993.

[31] YU J. and XIANG Y., ‘‘Hypermedia Presentation and authoring System’’, *Proceedings of the sixth international World Wide Web,* 7–11 Avril, Santa Clara, USA.

[32] ZIZI M., BEAUDOIN–LAFON M., ‘‘Accessing Hyperdocument through Interactive Dynamic Maps’’, *ECHT ’94 Proceedings*, pp. 126–134, September 1994.