Institutional Repository - Research Portal
Dépôt Institutionnel - Portail de la Recherche
researchportal.unamur.be
University of Namur

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Visualisation et édition temporelle dans Madeus

Meurisse, Miguel

Award date:
1999

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 02. May. 2026

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX, NAMUR

INSTITUT D'INFORMATIQUE

RUE GRANDGAGNAGE, 21, B-5000 NAMUR (BELGIUM)

Année Académique 1998-1999

Visualisation et édition temporelle dans MADEUS.

Miguel Meurisse

Mémoire présenté en vue de l'obtention du grade de

Maître en Informatique

Remerciements

Je remericie tout d'abord Laurent Tardif pour ses conseils pertinents et sa grande disponibilité en tout moment.

Je souhaite aussi remercier toute l'équipe du projet OPERA de l'INRIA Rhône Alpes. Que se soit Vincent, Cécile, Murielle, Nabil, Irène, Laurent(s), Lionnel, Fréderic(s), Stéphane, Yves, Ramzi, Marrion, Loay ou José leur disponibilité et leur symphatie ont fait de se stage une expérience unique.

Je tiens également à remercier Laurent Kempeneers et Benjamin Leonard pour avoir établis les contacts nécessaire à la préparation du stage.

Je remerice messieurs Fichefet et Leclerc d'être les promoteurs de ce mémoire.

Je remerice tout le monde m'ayant encouragé et aidé à réaliser ce travail.

Enfin je remercie les membres du jury de consacrer une partie de leur temps pour lire ce document :-).

Résumé

MADEUS est un éditeur de documents multimédia basé sur une approche relationnelle. Les différentes relations sont matérialisées par des contraintes. Cette approche permet à l'auteur de spécifier un scénario facilement sans se préoccuper de la spécification des instants de début et de fin des médias le composant. En effet, le système analyse le jeu de contraintes temporelles insérées dans le document et fournit une solution particulière afin de spécifier l'ordonnancement temporel de la présentation. Cette façon de procéder est très pratique, mais impose une phase de test de cohérence et de formatage assez lourde. Dans cet éditeur, nous nous sommes intéressés à la composante temporelle. Le but de ce mémoire est de permettre une première approche de l'édition temporelle directe dans une vue graphique dédiée. Nous nous sommes limités pour ces premiers pas à une modification de la durée d'un média appartenant à la présentation en cours de création. Pour ce faire, nous devons tout d'abord afficher le scénario temporel du document dans la vue graphique. Afin de rendre l'édition plus intuitive, nous proposons aussi quelques méthodes permettant d'améliorer l'affichage à cet effet. Les documents étant représentés en interne par un graphe temporel, nous proposerons donc une façon de visualiser un graphe en tenant compte de critères spécifiques à l'édition temporelle. La méthode d'édition de la durée d'un objet présentée ici est basée sur une modification d'une valeur préférée contenue dans un intervalle. Elle doit garantir que cette nouvelle durée exprimée ne modife pas le jeu de contraintes imposé par l'auteur.

Abstract

MADEUS is a multimedia document editor based on a relational approach. The relations are materialized by constraints. This approach allows to specify easyly a temporal scenario without being concerned with specification of the beginning and ending moment of the medias. Indeed, the system analyzes the temporal constraints'set inserted in the document and provides a particular solution in order to specify the temporal scheduling of the presentation. This way of proceeding is very practical, but imposes a coherency testing and a formatting phase. In this authoring tools, we were interested in the temporal component. The goal of this thesis is to allow a first approach of the direct temporal edition in a dedicated graphical view. We limited ourselves for these first steps to a modification of media's duration during the edition. With this intention, we must, first of all, display the temporal scenario of the document in the graphical view. In order to make the edition more intuitive, we propose also some methods allowing to improve the display. The internal document's representation is made by a temporal graph, we will thus propose a way of visualizing a graph by taking account of criteria specific to the temporal edition. The edition's method of an object's duration presented here is based on a modification of the preferred value contained in the interval. It must also guarantee that this new expressed duration did not modify the constraints'set imposed by the author.

Table des matières

1 Introduction **13**
- 1.1 Définition d'un document multimédia . . . 13
- 1.2 Edition de documents multimédias . . . 13
- 1.3 Composante temporelle d'un document MADEUS . . . 14
- 1.4 objectif du stage . . . 15
- 1.5 Présentation du travail . . . 15

2 Edition de document multimédia **17**
- 2.1 Introduction . . . 17
- 2.2 Classification des outils multimédias . . . 17
  - 2.2.1 Les logiciels "classiques" . . . 17
  - 2.2.2 Les langages de programmation . . . 18
  - 2.2.3 Les systèmes auteurs . . . 18
- 2.3 Classification des systèmes auteurs . . . 18
- 2.4 Approche opérationnelle . . . 19
  - 2.4.1 Les langages de scripts . . . 19
  - 2.4.2 Placement temporel absolu . . . 21
- 2.5 Approche relationnelle . . . 23
  - 2.5.1 SMIL . . . 24
  - 2.5.2 Approche par contrainte . . . 26
- 2.6 Conclusion . . . 30

3 MADEUS **33**
- 3.1 Introduction . . . 33
- 3.2 La vue hiérarchique . . . 33
- 3.3 La vue d'exécution . . . 34
  - 3.3.1 Informations dynamiques . . . 34
  - 3.3.2 Informations statiques . . . 34
- 3.4 La vue Time Line . . . 35
  - 3.4.1 Généralités . . . 35
  - 3.4.2 Affichage des contraintes . . . 37
  - 3.4.3 Affichage du document . . . 38
  - 3.4.4 Interactions de l'utilisateur . . . 38

3.5 Synchronisation des vues 38
3.6 Désynchronisation des vues 39
3.7 Conclusion 40

4 Edition dans MADEUS **41**
4.1 Introduction 41
4.2 Edition de la structure 41
4.2.1 Ajout d'un objet 42
4.2.2 Suppression d'un objet 42
4.2.3 Déplacement d'un objet 42
4.2.4 Remarque 42
4.3 Edition spatiale 43
4.3.1 Changement de la position d'un objet 43
4.3.2 Edition des contraintes spatiales 43
4.4 Edition temporelle dans la vue Time Line 43
4.4.1 Remarques préliminaires 43
4.4.2 Selection du média 44
4.4.3 détérmination de la structure solidaire de l'objet. 44
4.4.4 Calcul de l'intervalle de retaillage possible 45
4.4.5 Edition proprement dite 50
4.4.6 Mise à jour des coordonnées d'affichage 51
4.5 Edition temporelle dans les autres vues 52
4.5.1 Préparation de l'édition 52
4.5.2 Détermination de l'intervalle de retaillage 52
4.5.3 Affichage de l'intervalle 53
4.5.4 Edition proprement dite 54
4.5.5 Modification de l'affichage 54
4.6 Remarque finale sur l'édition temporelle 55
4.7 Conclusion 55

5 Visualisation de graphes **57**
5.1 Introdocution 57
5.2 Caractéristiques d'un graphe 58
5.3 Dessin de graphe 58
5.3.1 Polyline drawing 58
5.3.2 Straight-line drawing 59
5.3.3 Orthogonal drawing 59
5.3.4 Planar drawing 59
5.3.5 Hierarchical drawing 60
5.3.6 Upwards drawing 61
5.3.7 Combinaison de méthodes 61
5.4 MADEUS et le dessin de graphe 62
5.4.1 Découpe du graphe en niveaux 62

5.4.2 Arrangement des noeuds dans un même niveau 62
5.4.3 Assignation des coordonnées 63
5.5 Conclusion 64

6 Visualiation du graphe temporel dans la vue Time Line 67
6.1 Introduction 67
6.2 Données utiles 68
6.2.1 Copie du document original 68
6.2.2 Ajout de l'information temporelle 68
6.2.3 Récupération des coordonnées d'affichage 68
6.3 Simple Affichage du document 69
6.4 Affichage plus compact du document 70
6.4.1 Détermination du degré maximun de liberté par rapport aux lignes supérieures 71
6.4.2 Ajustement du degré de liberté par rapport aux lignes supérieures 71
6.4.3 Détermination du degré de liberté par rapport à la première ligne 72
6.4.4 Modificataion des données dans le document 72
6.4.5 Modification de l'affichage 72
6.5 Affichage du document minimisant les croisements 73
6.5.1 Découpe en niveau du graphe 73
6.5.2 Recherche des croisements 73
6.5.3 Détermination de la racine 74
6.5.4 Détermination de la partie à bouger 75
6.5.5 Mise à jour des coordonnées 75
6.6 Affichage compact minimisant les croisements 76
6.7 Conclusion 77

7 Perspectives 79
7.1 Introduction 79
7.2 Placement personnalisé des objets dans la vue Time Line 79
7.2.1 Sélection du média à bouger 80
7.2.2 Limitation du déplacement 80
7.2.3 Modification de son ordonnée 80
7.2.4 Mise à jour des coordonnées d'affichage 81
7.2.5 Remarque 81
7.3 Visualisation des hyperliens 81
7.3.1 Liens Intra-Documents 82
7.3.2 Liens inter-Documents 84
7.4 Edition des contraintes temporelles dans la vue Time Line 84
7.4.1 Ajout d'une contrainte 84
7.4.2 Suppression d'une contrainte 86

7.5 Déplacement d'un objet dans le scenario temporel 87

8 Conclusion **89**
8.1 Synthèse 89
8.2 Perspectives 90

Bibliographie **93**

Table des figures

2.1 le système auteur iSHELL 21
2.2 le système auteur DIRECTOR 23
2.3 le système auteur GRiNS 26
2.4 Correspondance entre les opérateurs d'Allen et les relations de HPAS 28
2.5 Vue d'édition temporelle de HPAS 29
2.6 le système auteur MADEUS 31

3.1 Conventions de représentation des médias 36
3.2 Conventions de représentation des delais 36
3.3 Visualisation particulière d'une infinité de solutions 36
3.4 Visualisation des contraintes d'Allen 37
3.5 Modification de l'affichage suite à une ouverture/fermeture d'un composite 39

4.1 différentes formes du curseur 44
4.2 vérification de chemin 47
4.3 délais empéchants le retaillage 49
4.4 délais à droite limitant l'augmentation de la taille 49
4.5 délais à gauche limitant l'augmentation de la taille 50
4.6 Première possibilité d'affichage de l'intervalle 53
4.7 Deuxième possibilité d'affichage de l'intervalle 54
4.8 Possibilité d'édition de l'intervalle non retenue 55

5.1 Problème d'un bon placement de noeuds dans le plan 57
5.2 Création d'une représentation d'Orthogonal drawing 60
5.3 Réprésentation d'un graphe en Upwards drawing 61
5.4 Ordonnancement des noeuds dans un niveau 63
5.5 Méthode du pendule 64

6.1 affichage temporel dans la vue Time Line 70
6.2 différentes étapes du "compactage" de l'affichage 72
6.3 Croisements entre médias et relations dans la vue Time Line . . . 74
6.4 Etapes trois et quatre de l'élimination des croisements 76
6.5 Modificaton des coordonnées des divers éléments 77

7.1 Différentes formes du curseur 83
7.2 Convention de représentation des liens intra-documents 83
7.3 Convention de représentation des liens inter-documents 84
7.4 Suppression d'une contrainte avec duplication de noeud 86
7.5 Suppression d'une contrainte sans duplication de noeud 87

Chapitre 1

Introduction

1.1 Définition d'un document multimédia

De nous jours, le concept de document a évolué. En effet, le document classique évolue et l'on commence à voir une généralisaton de documents multimédia. Cette mutation est surtout due à l'évolution des configurations informatiques permettant une gestion parfaite de ces nouveaux documents et du développement exponentiel d'Internet qui propose de plus en plus de sites intégrant de nouveaux médias.

Un document multimédia se différencie d'un document classique à cause de la nature des objets manipulés dans celui-ci. En effet, le multimédia permet une inclusion de sons ou bien de vidéos dans un document. L'intégration de tels objets demande la prise en compte d'un facteur nouveau : le temps. Cette composante temporelle s'ajoute donc aux trois autres composantes d'un document classique qui rappelons-le sont les suivantes :

- dimension spatiale qui régit la présentation graphique du document.
- dimension logique qui s'occupe de la structuration et de la décomposition de celui-ci en entités propres (par exemple un chapitre dans un livre)
- dimension sémantique qui permet une navigation dans le document ou bien de document à document.

L'apparition de cette nouvelle demande de documents amène bien évidemment un besoin de création d'outils permettant la visualisation d'un part et l'édition d'autre part de ces données multimédia.

1.2 Edition de documents multimédias

La majorité des éditeurs existant sur le marché se base sur une approche de création de documents régie par un langage de script. La ressemblance de ce

procédé avec l'utilisation d'un langage de programmation rebute certaines personnes à cause d'une appréhension de difficultés insurmontables. C'est pourquoi le projet OPERA de L'INRIA Rhône Alpes essaie depuis quelques années de créer un éditeur muni d'une interface graphique intuitif et agréable permettant à n'importe quel utilisateur (que se soit un expert ou bien un utilisateur occasionnel) de définir un document multimédia. Le prototype en cours de développement se nomme MADEUS.

Une édition de qualité est seulement possible si l'auteur arrive à maîtriser l'agencement et le déroulement de la présentation qu'il est en train de réaliser. Pour ce faire, une visualisation des quatre composantes pour chaque objet utilisé est souhaitable. Meilleur sera l'affichage de ces informations, meilleur sera la motivation de l'auteur de travailler avec l'outil.

Les documents traités par MADEUS reprennent bien évidemment les quatre composantes citées ci-dessus. Une attention particulière dans la conception de l'éditeur permet pour chaque objet appartenant au scénario de la présentation de connaître l'état de ces composantes à tout moment du processus d'édition. Dans ce travail, nous nous concentrerons sur la composante temporelle des documents.

1.3 Composante temporelle d'un document MADEUS

La création du scénario temporel de MADEUS est basée sur une approche relationnelle. En effet, l'auteur définit les médias à utiliser et les assemble grâce à un lot de contraintes disponibles. Par exemple, il peut vouloir exprimer qu'une musique introductive sera présentée avant une vidéo. Le scénario de ce document incluera donc une contrainte du genre : musique BEFORE vidéo.

Cette approche par contraintes est intéressante. Elle permet de spécifier un agencement temporel sans que l'auteur doive se préocupper d'un placement quelconque de ces objets sur une ligne du temps. En effet, le système avant de présenter la composition de l'auteur calculera les instants de début et de fin de présentation des différents médias et cal automatiquement. Ce procédé permet de débarrasser le créateur du document d'une tâche fastidieuse et parfois très complexe.

1.4 objectif du stage

Une bonne visualisation du scénario temporel est indispensable pour effectuer des mapulitations directes de celui-ci dans une vue graphique. Celle-ci masque la complexité des opérations internes de mise à jour du document. Dans le développement de MADEUS une grosse analyse a été effectuée pour définir les besoins de visualisation adaptés à la présentation du scénario temporel. Le but du stage était d'une part de proposer une manière d'effectuer l'affichage initial de la composante temporelle pour permettre une intégration des travaux réalisés précédemment et d'autre part de commencer la phase d'édition temporelle directe dans la vue graphique créée lors de la mise en oeuvre de la première partie. Cette entrée en matière présentera la modification de la durée d'un média.

Comme nous l'avons déjà souligné, la visualisation optimale des données étant un souci pour l'édition dans un environnement graphique, une autre partie s'est donc implicitement ajoutée. Il s'agit d'obtenir un affichage à l'écran le mieux adapté aux besoins de l'édition temporelle du document.

1.5 Présentation du travail

Dans ce mémoire, nous allons donc nous concentrer sur les deux objectifs cités à la section précédente. Pour ce faire, nous commencerons dans le chapitre 2 par effectuer un état de l'art de la visualisaton temporelle dans différents systèmes auteurs.
Une fois cette analyse effectuée, nous regarderons dans le chapitre 3 ce que propose MADEUS en ce qui concerne ces facilités de visualisation des quatre composantes d'un média dans le document.
Au chapitre 4, nous aborderons les outils d'éditions proposés en nous attardant sur l'édition directe de la durée d'un objet dans la vue dédiée à cette tâche. Le jeu de contraintes limitant les désirs de l'auteur, nous présenterons une méthode garantissant l'intégrité du scénario.
La suite de l'exposé sera consacrée à la mise en place de l'affichage initial. La structure interne représentant le scénario du document étant un graphe, nous poursuivrons au chapitre 5 par un état de l'art de la visualisaton de ces structures pour terminer au chapitre 6 par l'explication des différentes étapes constituant la solution choisie. En plus d'une simple visualisation du graphe temporel, nous allons améliorer sa présentation dans le but de rendre l'édition plus efficace.

Chapitre 2

Edition de document multimédia

2.1 Introduction

Dans ce chapitre, nous nous concentrerons sur une partie des différents outils mis à notre portée pour concevoir un document multimédia. Il existe une pléthore d'éditeurs sur le marché (rien que pour les systèmes auteurs, le site web [2] en référence plus de 70 !), c'est pourquoi nous allons simplement dans la suite énumérer un certain nombre de catégories d'outils et pour chacune d'elles en présenter un en particulier qui aura retenu notre attention.

2.2 Classification des outils multimédias

Dans son travail, Luc Goffinet [13] nous propose une approche se déroulant en deux temps. Il conçoit la conception d'un document multimédia en commençant par récupérer les différents composants utiles à la présentation. La création ou bien la retouche de ceux-ci s'effectue dans des outils dit "monomédia". Voici une liste des pricipaux traitements qu'il est possible de trouver lors de cette phase de préparation

- traitement des images
- traitement des sons
- traitement des vidéos
- traitement d'objets animés
- outils de gestion de format

La seconde phase quant-à-elle intègre les données sur lesquelles on a travaillé à l'étape précédente. Suivant la complexité de la présentation à réaliser, on peut compter trois catégories d'outils pouvant aider l'auteur.

2.2.1 Les logiciels "classiques"

Ce sont principalement les logiciels de bureautique standard.

Par exemple, de nos jours, il est tout à fait conventionnel dans des traitements de texte évolués de pouvoir insérer en plus du texte, des images et même des annotations vocales. De plus, les logiciels de présentations intégrées dans les différentes suites commerciales permettent aussi quant-a eux de créer des "slide-show" très facilement.

2.2.2 Les langages de programmation

Cette catégorie d'outil est bien sûr à réserver lorqu'il n'est pas possible de réaliser un projet avec les outils des autres catégories. C'est une approche intéressante, lorsqu'il faut garantir une grande vitesse de presentation non toujours retrouvée dans un code interprété. En effet, avec un langage de programmation, on peut optimiser les capacités hardware et software d'une machine propre. De plus la portabilité du code de certains langages (comme par exemple JAVA) permet une migration de plate-forme à moindre frais. Malgré tous ces avantages, cette optique est de moins en moins utilisée à cause de la complexité de réalisation que procure cette approche.

2.2.3 Les systèmes auteurs

Un système auteur est un programme d'arrangement de médias (il peut aussi intégrer des outils de traitement vus à la première phase). Cet arrangement est piloté par l'interface du programme et celui-ci propose aussi des instruments afin de faciliter la présentation et l'interaction dans le document.

Le but des systèmes auteurs est de permettre à des personnes n'étant pas des informaticiens chevronnés de construire des documents multimédias assez complexes sans écrire une ligne de code. L'avantage principal de cette catégorie d'outils est sa capacité à fournir un développement beaucoup plus rapide que celle proposée par les langages de programmation. Qui dit développement plus rapide, dit aussi réduction des coûts.

La suite du chapitre sera entièrement consacrée à ce type d'outils.

2.3 Classification des systèmes auteurs

Ces outils étant les plus attrayants, nous avons décidé d'y regarder de plus près. Pour essayer de s'y retrouver dans la multitude de systèmes existants, nous devons les classifier. Pour ce faire, nous allons nous baser sur la classification effectuée par Murielle Jourdan, Nabil Layaïlda et Cécile Roisin dans [19]. Ils discernent entre autre deux grandes approches du problème :

l'approche opérationnelle

L'auteur détermine lui-même la durée des différents objets du document. Il doit aussi s'occuper du placement temporel de ceux-ci. En résumé, il doit fixer les instants de début et de fin de tous les objets intervenant dans la présentation.

l'approche relationnelle

Dans cette approche, l'auteur spécifie seulement les relations existantes entre les différents médias composant le document. Ce sont donc les relations qui régissent l'enchaînement temporel. Le système auteur quant-à lui calcule lui même les durées de présentations des différents médias contenus dans le document.

Dans la suite, nous allons caractériser et affiner cette découpe.

2.4 Approche opérationnelle

Dans cette section, nous allons énoncer différents courants existants pour aider l'auteur à spécifier les données temporelles de son document. Pour chacun d'entre eux, nous présenterons un outil particulier. Dans cette analyse, nous nous concentrerons principalement sur la manière proposée à l'auteur pour créer son document et sur la visualisation des informations temporelles dans le système auteur. Les deux courants présentés sont les suivants

- Les langages de scripts
- Le placement temporel absolu

2.4.1 Les langages de scripts

Cette manière de procéder est la plus répandue de nos jours. Les outils de cette catégorie proposent à l'auteur beaucoup de fonctions prédifinies pour modeler son document comme il l'entend. Elles permettent de spécifier complètement la présentation que se soit du point de vue temporel ou bien au niveau de l'interactivité d'un objet. Le document est en fait une suite d'enchaînements d'invocations de commande qui forment le script.

Pour illustrer cette approche, nous avons choisi de présenter iSHELL. Notre choix a été motivé par le fait que ce logiciel est gratuit et téléchargeable directement sur le site web de Tribeworks [17]. C'est une bonne opportunité d'analyser le comportement d'un programme qui, du fait de sa distribution libre, peut aussi bien être utilisé par des personnes non expertes que par des professionnels de la discipline.

Création du scénario temporel

la création du scénario temporel d'iSHELL est basé sur un modèle événementiel [18]. L'association d'actions à un événement se produisant permet de construire

le scénario de la présentation. Les deux types d'événements principaux sont le début de présentation d'un objet et la disparition de celui-ci de l'écran. Les actions temporelles déclenchables sont les suivantes :

- démarrer la présentation d'un objet.
- arrêter la présentation de cet objet.
- faire une pause dans la présentation.
- modifier le débit de présentation.
- spécifier un instant de début de présentation X unités de temps après l'apparition de l'événement déclencheur.

Vu cet ensemble d'actions, la construction classique du scénario se produit en trois grandes étapes :

1. Définition des objets présents dans le document
2. Activation des événements à contrôler objet par objet.
3. Assignation des actions à un événement donné. Ces actions se réalisent si et seulement si l'événement se produit. On remarque bien que les actions associées à un événement de fin de présentation d'un média seront le démarrage de la présentation des objets le suivant dans le scénario temporel. Les actions associées aux événements de début de présentation serviront surtout pour gérer l'interactivité du document

Visualisation et édition du scénario

Un des grands désavantages de ce programme est de ne pas posséder de vue dédiée à la visualisation d'informations temporelles. La seule manière de la connaître est de se balader dans la vue structurale du document. Tous les éléments sont représentés par une structure hiérarchique. Un des noeuds du premier niveau de cette structure nous informe des événements à vérifier. Un sous-niveau nous présente les actions associées à celui-ci. Du point de vue édtion, un "grag and drop" de la fenêtre d'édition vers l'événement cible permet de lui assigner une action qui sera pauffinée directement dans la vue structurale.

Evaluation de l'outil

L'outil est assez facile à utiliser. L'utilisation du "glisser déplacer" permet des assignations rapides et précises. La vue structurale est bien pensée, il est possible d'ouvrir ou bien de fermer n'importe quel niveau de hiérarchie pour un objet donné. Pour ne pas devoir se servir trop souvent des ascenceurs dans cette vue, un mécanisme permet d'ouvrir une fenêtre supplémentaire ne contenant que la partie de la hiérarchie se trouvant en dessous du noeud sélectionné. Grâce à cette fonctionnalité, en ouvrant un seul niveau l'auteur possède une vision globale des objets contenus dans la présentation.

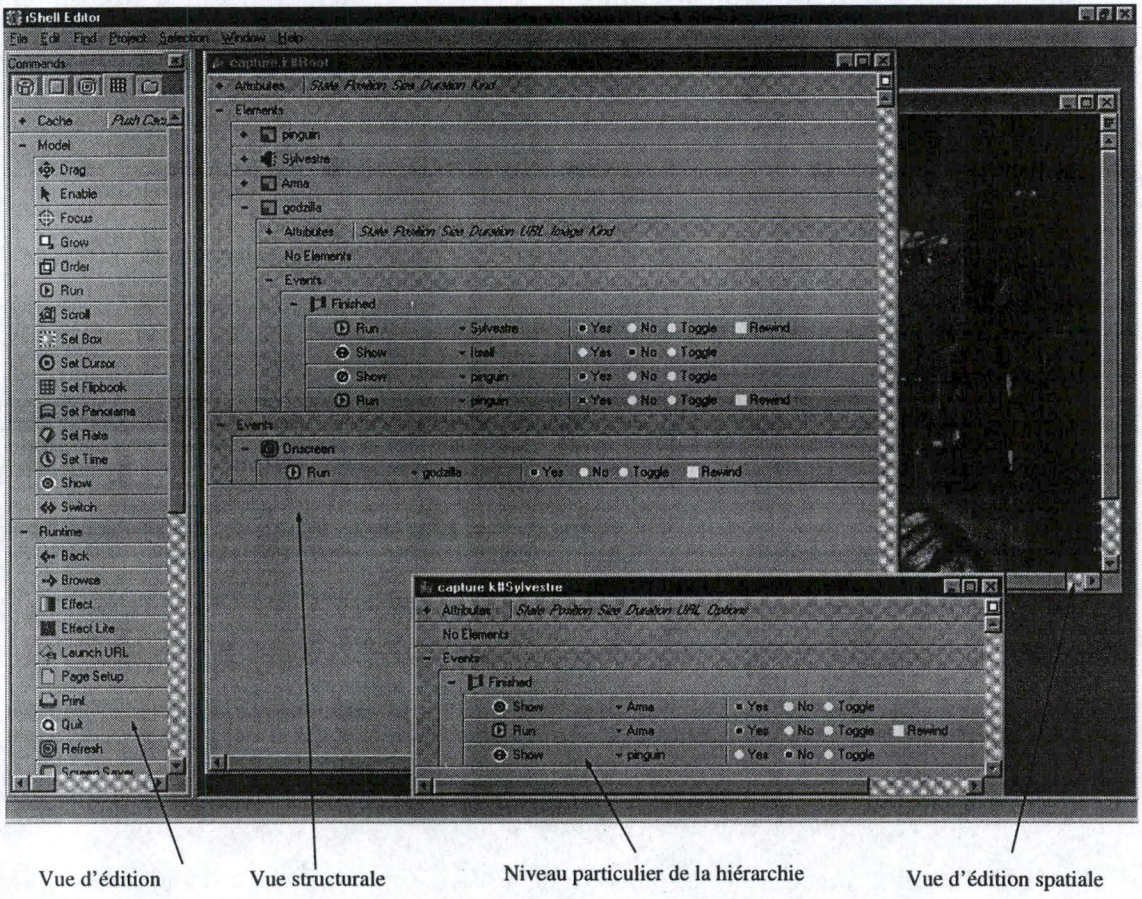

FIG. 2.1 – *le système auteur iSHELL*

Par contre le manque d'une vue dédiée à l'information temporelle est omniprésent. L'auteur à très difficile de se représenter la présentation en construction. Si une erreur à été commise lors de la phase d'assemblage des médias, il est parfois difficile de retrouver les informations voulues dans la structure et de les déplacer à sa guise. De ce fait, l'aspect d'une conception incrémantale du document n'est pas évidente lorsque le scénario devient conséquent.

2.4.2 Placement temporel absolu

Dans cette approche, une vue spéciale est dédiée aux informations temporelles du document. Cette vue présente tout le scénario sur une ligne du temps et met à la disposition des utilisateurs un nombre de pistes permettant de déposer les éléments de la présentation. De ce fait, l'auteur peut assigner très facilement les moments de début et de fin des médias présentés. Le problème principal qui se pose avec cette façon de travailler est le manque implicite de traitement de l'in-

terraction. C'est pourquoi cette approche est souvent combinée avec un langage de scripts.

L'outil que nous avons choisi de présenter combine ces deux approches, il s'agit de DIRECTOR [27]. Notre choix à été motivé par le fait que ce logiciel est le plus utilisé du marché. De plus les professionnels lui faisant confiance, nous avons donc cru bon de l'analyser.

Création du scénario temporel

La création du scénario temporel se deroulé en trois étapes [8].

- Définition des objets de base du document. DIRECTOR est le seul logiciel présenté ici qui permet une série de traitements "monomédia". De ce fait cette étape peut être une simple intégraton de média ou bien un acte de création. Ces différents éléments sont répertoriés dans une fenêtre spéciale appelée le "cast".
- Determination de la durée des objets et placement de ceux-ci sur les différentes pistes de la vue Time Line
- Détermination de l'interactivité du document. Pour ce faire DIRECTOR utilise un langage de scripts nommé LINGO.

Visualisation et édition du scénario

La visualisation de l'info temporelle est regroupée dans la vue Time Line. Celle-ci intègre les différents médias. On peut identifier facilement les instants de début et de fin de chaque objet. Des pistes spéciales représentent l'interactivité du document. Sur celle-ci, on peut par exemple visualiser la période de validité des boutons inclus dans la présentation. L'interface de DIRECTOR est présenté à la figure 2.2.

Evaluation de l'outil

DIRECTOR est un outil de création multimedia intégré. Si on n'a que cet outil sous la main, il est capable de tout. La vue Time Line donne une très bonne approche de la visualisation du scénario temporel. L'insertion d'un story board facilite la création de scènes et leur intégration. Le cast est très pratique pour la reutilisabilité d'objet. Une fois ceux-ci insérés dans cette vue, les recuperer prend un temps minimun.

Par contre le grand problème de l'approche du placement temporel absolu est la rigidité du document. C'est-à-dire qu'il est très difficile d'insérer un objet en plein milieu du scénario. Et de ce fait, il faut bouger temporellement tous les médias présentés après l'insertion. Cette pratique implique obligatoirement un

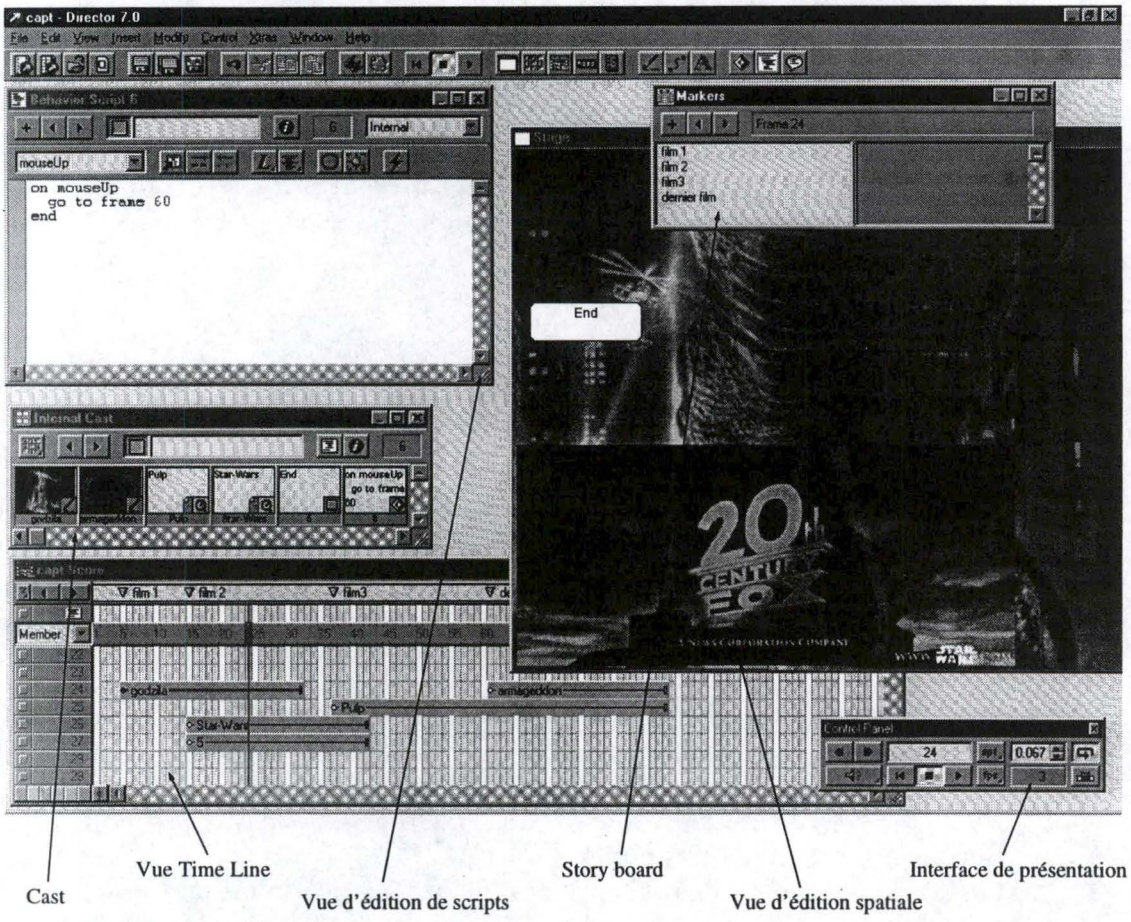

FIG. 2.2 – *le système auteur DIRECTOR*

design de la présentation incluant une fort grande partie de conception de l'enchaînement du document avant de pouvoir commencer réellement la réalisation. Il existe aussi un problème avec la spécification des durées donnée à un objet incontrôlable. Il se peut que le temps de présentation total soit différent de celui donné par les estimations de la Time Line.

2.5 Approche relationnelle

Dans cette approche, l'auteur décrit des relations entre les objets présents dans le doucment. Par rapport à l'approche opérationnelle, cette façon de travailler permet une structuration et une gestion du scénario beaucoup plus importante. De ce fait, des changements incrémentaux se résument à une modification des relations entre les objets mis en jeu. Il existe deux grands courants que nous allons découvrir dans la suite : SMIL et l'approche par contraintes.

2.5.1 SMIL

Pour illustrer cette approche, nous avons choisi de présenter GRiNS [14]. C'est un éditeur de documents multimédias basé sur SMIL (Syncronized Multimedia Intergration Language) [34]. Par ses caractéristques intrinsèques, ce langage permet de spécifer le document comme une structure d'arbre. Dans celle-ci chaque noeud est associé à un opérateur temporel tandis que chaque arc matérialisé un objet.

Création du scénario temporel

La création du scénario temporel est permise d'une façon incrémentale. C'est-à-dire que chaque fois que l'on spécifie un opérateur pour un noeud, on peut en même temps assigner les objets associés à celui-ci. Cette approche est intuitive et conforme à la structure d'un document SMIL. Les différents opérateurs temporels disponibles sont les suivants :

- PAR : La présentation de tous les objets spécifiés pour ce noeud démarre en même temps. On parle alors de présentation parallèle des médias.
- SEQ : La présentation des objets se deroule séquentiellement. L'instant de fin d'un objet est l'instant de début de l'élément suivant de la séquence.

SMIL permet cetaines formes de syncronisation temporelle :

- Spécification de l'instant de fin d'un PAR
  On peut spécifier explicitement la fin de la présentation d'un PAR. Pour ce faire, il faut assigner à l'attribut endsync du PAR les valeurs suivantes :

  first : dans ce cas, la durée du PAR sera la plus petite durée spécifiée pour un de ces éléments

  last : dans ce cas, la durée du PAR sera la plus grande durée spécifiée pour un de ses éléments. C'est le comportement normal d'un PAR.

  id-ref: dans ce cas, la durée du PAR sera le durée spécifiée pour l'objet identifié par la référence id-ref.

- Synchronisation entre les médias.
  Il est possible de synchroniser plusieurs objets ensemble. Pour ce faire, nous devons ajouter d'une part l'identifiant de l'objet sur lequel on doit se synchroniser et d'autre part le type de synchronisation qui peut être

  begin : synchronisation des deux instants de début des objets

  end : présentation séquentielle des deux objets. L'objet synchronisant étant présenté le premier.

  clock-value : Le début de la présentation de l'élément synchronisé commence après le temps spécifié comme clock-value du début de la présentation de l'autre objet.

Visualisation et édition du scénario

Comme expliqué dans [15], la visualisation du scénario temporel se fait grâce à une vue Time Line. Celle-ci représente tous les objets présentés dans le document. Elle est divisée en lignes. Chaque ligne représente une région du document. La synchronisaton entre objets est référencée par des flèches à l'écran. On peut aussi remarquer une ligne du temps dans le bas de la vue. Celle-ci est dite "virtuelle". C'est-à-dire qu'elle donne seulement une indication du temps de la présentation. En effet, si il existe des objets incontrôlables dans celle-ci, on ne peut pas la deviner. Pour bien montrer ce phénomène, les objets dont on est pas sûr de la durée sont représentés différemmment dans la vue. GRiNS calcule lui-même le temps de présentation de ces objets. Cette vue ne sert pas à l'édition, elle sert seulement à visualiser la présentation.

Une autre vue, la vue structurelle nous permet quant-à-elle de voir la fameuse structure graphique. Les éléments impliqués dans un PAR se retrouvent l'un à coté de l'autre. Ceux impliqués dans un SEQ se retrouvent l'un en dessous de l'autre. Toutes les opérations d'éditions sont pilotées à partir de la vue structurelle. Les donnés non visibles directement dans la vue sont appelées grâce à une fenêtre de propriété associée à chaque objet et à chaque noeud de la présentation

Evaluation de l'outil

L'outil est très bien conçu. Le fait d'avoir deux vues principales pour la visualisation des informations temporelles est une bonne chose pour augmenter la lisibilité des informations. GRiNS se base sur SMIL qui est un langage élaboré par un organisme créateur de recommandations le W3C. Ceci lui procure un format utilisé qui n'est pas propriétaire. Si l'auteur veut changer d'éditeur, il n'aura pas trop de peine pour trouver un autre éditeur compatible avec ses documents existants.

Par contre, la visualisation dans GRiNS de gros documents n'est pas bien adaptée. En effet la représentation graphique de la structure prend une place considérable sur l'écran. Il en va de même pour la vue Time Line. Dans un cas où une multitude de régions sont définies dans le document, il devient assez difficile de s'y retrouver en un seul coup d'oeil. Pour finir, SMIL est un langage qui évolue. Une nouvelle recommandation est en cours d'élaboration [35], ce qui veut dire que le développement de l'éditeur pour respecter les nouvelles spécifications devra susciter beaucoup de nouvelles versions.

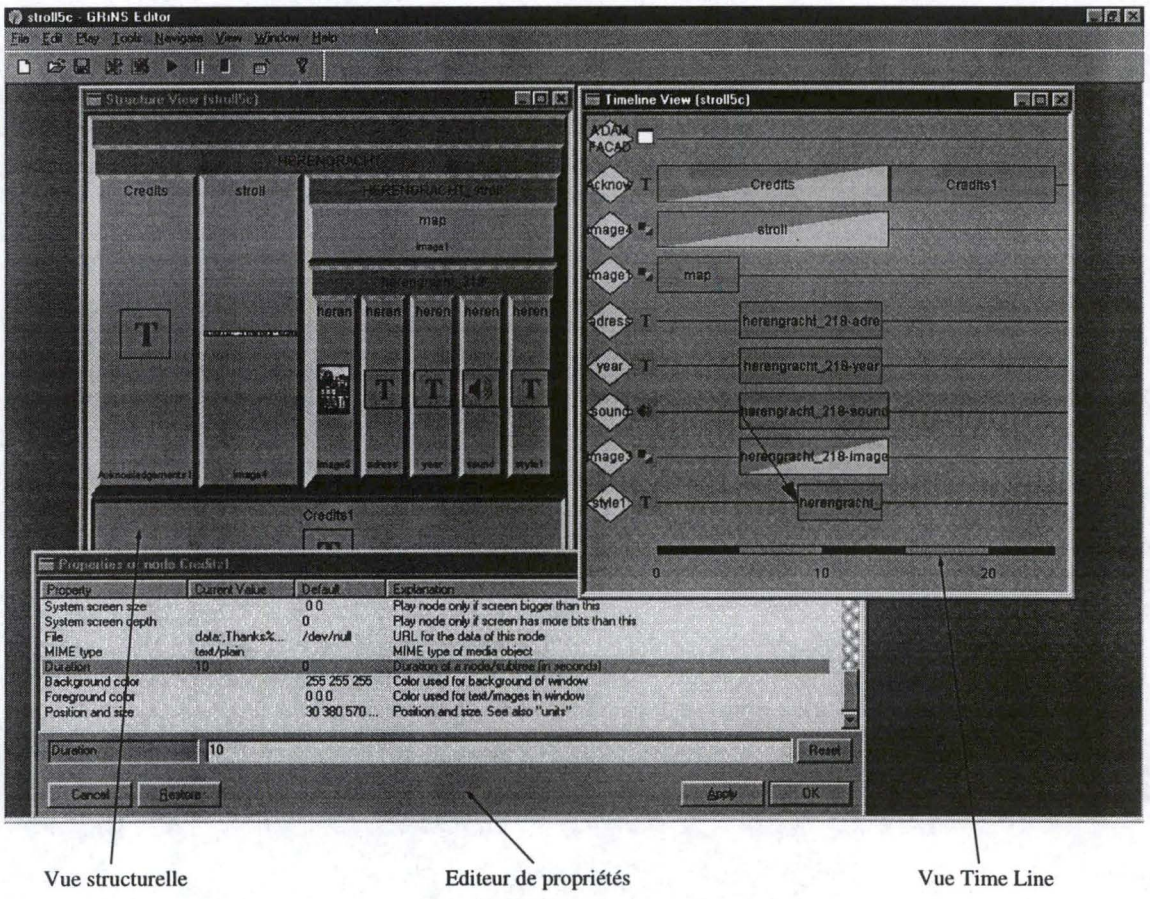

FIG. 2.3 – *le système auteur GRiNS*

2.5.2 Approche par contrainte

Cette approche permet de spécifier le scénario temporel du document en définissant un jeu de contraintes entre les différents éléments du document. Les contraintes utilisées peuvent se baser sur des instants de présentation ou bien sur un intervalle de présentation. Dans ce dernier cas, nous utiliserons les opérateurs d'Allen [1] pour spécifier notre scénario. L'emploi de contrainte ne permet pas à l'auteur de spécifier une durée fixe, mais c'est le formateur temporel qui va déterminer la durée de présentation optimale. Par conséquent, les informations concerant la durée seront exprimées sous la forme d'un intervalle caratérisé par

- une durée minimun
- une durée maximun
- une durée préférée

Une phase de test de la cohérence temporelle est obligatoire. A la fin de celle-ci, le document est déclaré cohérent si et seulement si il existe au moins une solution

particulière au jeu de contraintes exprimées dans le document. Si c'est le cas, le formateur temporel va choisi la durée de présentation des différents médias dans l'intervalle défini ci-dessus. Tant que possible, il essayera de choisir une valeur la plus proche de la durée préférée. Cette opération étant très complexe, nous ne nous intéresserons pas aux méthodes employées dans ce travail. Pour en savoir plus, il est intéressant de consulter [5] et [3].

Dans la suite de cette sous-section, nous allons nous intéresser à deux outils en particulier HPAS [36] et MADEUS [21]. Nous allons les analyser en utilisant le même schéma que celui proposé dans l'approche précedente.

HPAS

HAPS intègre une structure de données en arbre pour représenter le document [37]. Les noeuds de celui-ci représentent les médias à montrer et les arcs matérialisent les relations entre les médias. Il existe trois formes de relations :

- StartSync : L'instant de début de présentation des deux objets est le même.
- EndSync : L'instant de fin de présentation des deux objets est le même.
- SerialLink : L'instant de fin du premier objet est le même que l'instant de début du second.

Cet éditeur se basant sur les contraintes d'Allen, pour pouvoir les utiliser dans le scénario, l'auteur doit décomposer celles-ci pour en extraire les différentes relations citées ci-dessus. Une correspondance entre les deux est montrée à la figure 2.4.

Création du scénario temporel

La création du scénario se déroule en trois étapes :

- Créations des noeuds de la structure interne.

La suite du processus depend de l'approche de composition utilisée. On utilisera soit une approche "top-down" ou "bottom-up"

- "top-down" : Dans cette approche, l'auteur va dans un premier temps spécifier les relations incluses dans son document. Par la suite, il va seulement décrire les médias utilisés associés aux différents noeuds.
- "bottom-up" : Cette approche est l'inverse de la précedente. L'auteur spécifie ses médias avant de leur associer un jeu de relations matérialisant les contraintes d'Allen.

La partie interactivité est traitée lors de la phase de description des médias utilisés dans le document.

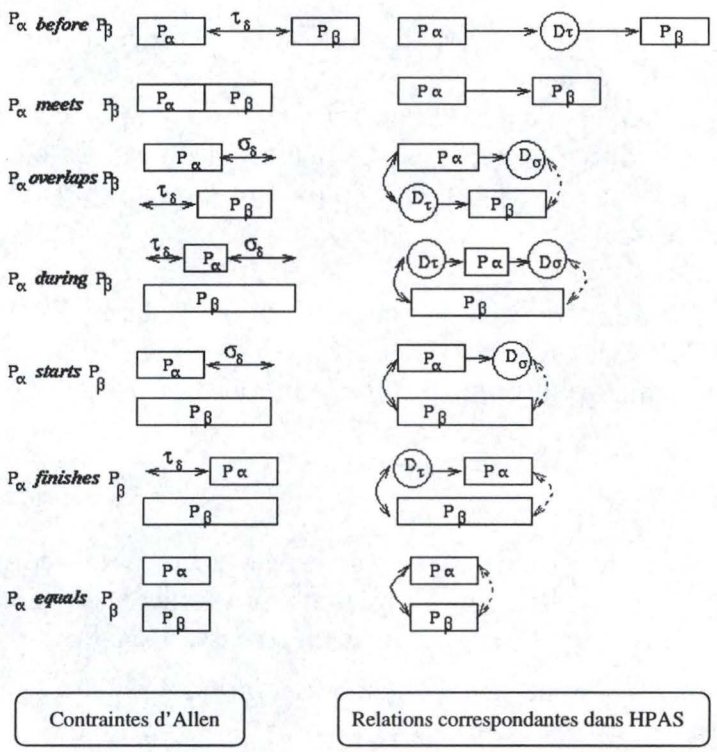

FIG. 2.4 – *Correspondance entre les opérateurs d'Allen et les relations de HPAS*

Visualisation et édition du scénario

La fenêtre d'édition temporelle de HPAS présente l'arbre associé au document. Les objets sont matérialisés suivant le choix de l'auteur par des icônes désignant leur type soit par leur identifiants. Les relations quant-à-elles suivent les conventions suivantes :

- StartSync : une double flèche dessinée en trait continu.
- EndSync : une double flèche dessinée en pointillées.
- SerialLink : une simple flèche en trait continu orientée vers l'objet à présenter en dernier.

On peut voir un exemple a la figure 2.5.

Evaluation de l'outil

L'approche basée sur les contraintes offre une flexibilité d'édition incrémentale fort agréable. La représentation des médias est assez basique, mais tout à fait suffisante pour avoir un bon aperçu des composants de la présentation.

Malheureusement, l'auteur peut avoir assez difficile lorsqu'il désire intégrer

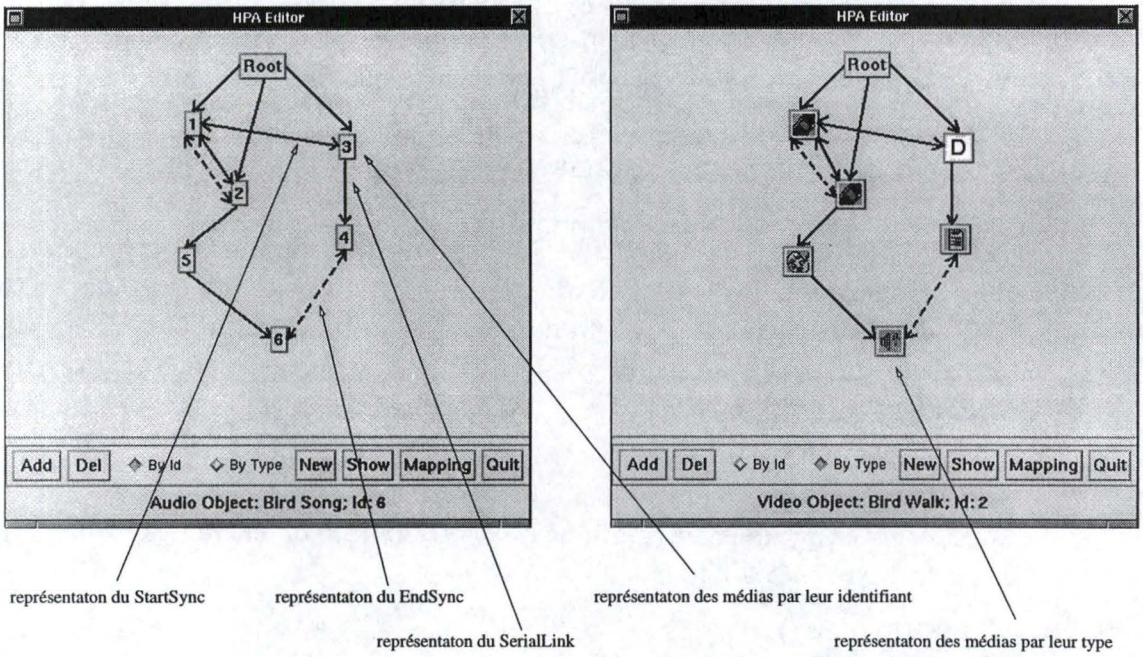

FIG. 2.5 – *Vue d'édition temporelle de HPAS*

une contrainte temporelle. En effet, il doit dans un premier temps imaginer la contranite d'Allen qu'il va employer, puis il doit la décortiquer en relations de bases pour ensuite les intégrer. Un autre risque dans un document comportant beaucoup de relation serait l'oubli de l'insertion d'une relations faisant partie d'une contrainte d'Allen. Dans ce cas, le deroulement du scénario ne serait pas conforme aux désidératats de l'auteur. L'erreur serait peut être difficile à détecter.

MADEUS

Dans cette sous-section, nous allons décrire sommairement l'outil MADEUS. La description détaillée se retrouvera tout au long de ce travail.

Nous retrouvons aussi dans MADEUS un positionnement spatial des objets à l'écran régi également par un jeu de contraintes.

Création du scénario temporel

La création du scénario temporel de MADEUS se base sur les opérateurs d'Allen. Trois autres opérateurs complètent les outils disponibles pour spécifier le scénario. Voici leur sémantique :

- A Parmax B : C'est la disjonction de A Starts B et de B Starts A
- A Parmin B : représente l'interruption de l'objet le plus court par l'objet le plus long.

- A Parmaster B : La durée de B est conditionnée par la durée de A. Si la durée de A est inférieure à celle de B, alors cela revient a un A parmin B. Tandis que si la durée de A est supérieure à celle de B cela revient dans ce cas à un B Starts A.

La composition du scénario se déroule en trois étapes :

- Creation de la structure hiérarchique :
  Dans MADEUS les documents sont organisés grâce à une structure hiérarchique. Pour définir cette structure, il existe des objets de type composite. Ces composites servant de containaire englobant d'autre objets. C'est l'imbrication multiple de ceux-ci qui permet la structuration hiérarchique du document.
- Caractérisation des médias
- Définition du jeu de contraintes
  Il est à noter qu'une contrainte ne peut être placée qu'entre deux objets de même niveau hiérarchique.

Visualisation et édition du scénario

La figure 2.6 nous montre un apperçu des différentes vues d'éditions. Chaque vue de MADEUS est dédiée à visualisation et à une édition d'informations particulières. Dans notre cas, la vue Time Line sera la vue de référence pour obtenir de l'information temporelle. Ceci ne veut pas dire qu'il est impossible de visualiser ou faire des modifications dans les autres vues. Nous pourrons nous en rendre compte dans le chaiptre 4.

Evaluation de l'outil

Comme son homologue HPAS, le système à base de contraintes permet une édition facile et une capacité de modifications du document assez aisée. La sémantique et surtout la forme des contraintes d'Allen étant respectées, l'auteur peut difficilement se tromper sur la transcription de son scénario.

Par contre, MADEUS étant en phase de développpement, il souffre de petits maux qui, espérons-le s'effacent petit à petit. Les travaux en cours permettront à terme de pouvoir bénéficier d'une édition directe dans les différentes vues ou bien encore d'une prise en compte de nouveaux formats de documents [20] par exemple.

2.6 Conclusion

En parcourant les différentes approches d'outils auteurs, nous avons pu effectuer certaines constatations sur la visualisation et l'édition de la composante

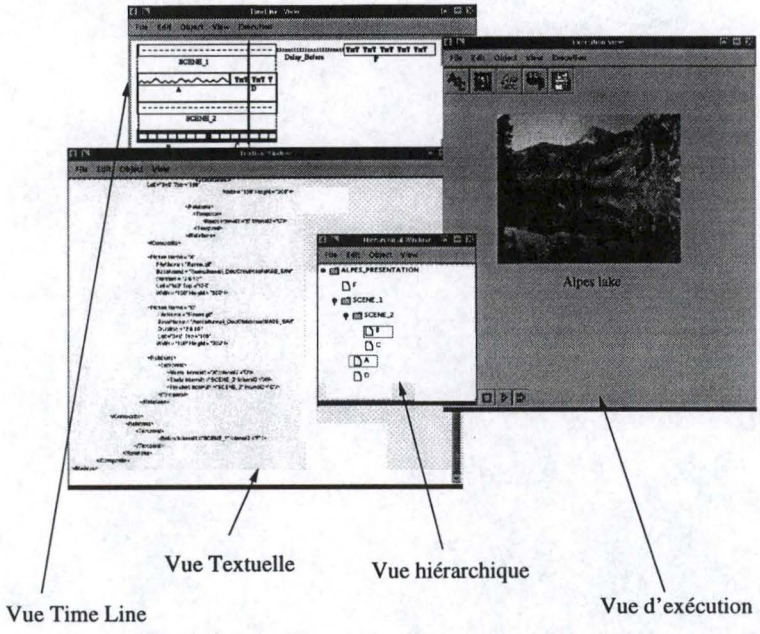

FIG. 2.6 – *le système auteur MADEUS*

temporelle. En voici un petit résumé :

- On remarque que malgré les différents systèmes présentés, ceux intégrants une vue time Line permettent de mieux cerner le scénario de la présentation.
- La représentation d'une vue Time Line doit être interprétée comme une aide à la visualisation du scénario plutôt qu'une copie conforme du résultat présenté à une unité de temps près. En effet, si certains objets incontrôlables sont présentés ou bien, si le document contient une part d'interactivité, alors les estimations de la Time Line seront faussées. C'est pourquoi il faut trouver un moyen de représenter le scénario tout en essayant de ne pas montrer de fausses informations.
- Un problème considérable dans la plupart des éditeurs est la gestion de la mise à l'échelle dans un gros document. Aucun d'entre eux ne propose de début de solution sauf iSHELL avec ses fenêtres représentant une partie de la hiérarchie. Cet outil se prête bien à la solution préconnisée puisqu'il ne montre pas directement l'information temporelle. Dans les autres, une telle pratique ne serait peut-être pas facile à mettre en oeuvre.
- L'édition directe d'un média dans une vue est beaucoup plus intuitive que la modification de propriétés appliquées à un objet.

Maintenant que nous avons relevé les points clés, nous allons dans les chapitres suivants voir si MADEUS satisfait aux attentes énoncées ci-dessus.

Chapitre 3

MADEUS

3.1 Introduction

Comme nous l'avons déjà souligné dans l'introduction générale, les documents multimédias traîtés par MADEUS intègrent les quatre dimensions suivantes :

- Dimension spatiale.
- Dimension hiérarchique.
- Dimension termporelle.
- Dimension hypermédia.

Une bonne compréhension d'un document MADEUS demande une excellente visualisation des quatre composantes citées ci-dessus. Une fois le document maîtrisé, toutes les opérations d'éditions se révèlent plus intuitives et plus claires. Par "plus intuitif", nous voulons exprimer le fait que l'auteur arrive d'une façon naturelle et très rapide à modifier une composante quelconque de son document. Une "édition claire" quant-à elle souligne la capacité de l'outil à fournir à l'auteur la meilleure représentation possible de l'exécution de sa composition.

En partant de cette constatation, nous en dérivons qu'un bon éditeur multimedia doit intégrer de la meilleure manière que ce soit la gestion de ces quatre dimensions. C'est pourquoi, dans ce chapitre, nous allons présenter les différentes possibilités de visualisation offertes par MADEUS pour un même document. Après une description succinte des dimensions hiérarchiques et spatiales, cet exposé, se concentrera principalement sur la dimension temporelle.

3.2 La vue hiérarchique

Dans cette vue, l'auteur du document peut analyser la structure de celui-ci. Nous savons que les documents produits par MADEUS peuvent avoir une structure hiérarchique, s'ils possèdent des composites. Dans cette vue, cette structure

se représente sous la forme de différents niveaux qui se distinguent par leur placement. Plus un niveau est à gauche, plus il est près de la racine de la hiérarchique. On peut aussi remarquer que les objets sont représentés par une icône précisant leurs statuts dans la hiérarchie (on emploiera une feuille pour désigner un média et un noeud pour désigner un composite). Ces illustrations sont suivies d'un identifiant (le nom donné par l'auteur à l'objet). L'auteur peut aussi interagir avec cette vue. Lorsque le document est chargé, la vue hiérarchique affiche seulement le premier niveau de la structure. Pour déplier celle-ci, il suffit de cliquer sur une icône représentant un composite. L'avantage de cette représentation est que l'on peut voir facilement la structure du document.Lorque nous avons affaire à des présentations fort complexes, l'espace d'affichage ne suffit plus pour voir toute la structure en un seul coup d'oeil. Pour remédier à cet inconveniant, nous pourrions nous intéresser aux travaux effectué par XEROX. Cette firme a developpé plusieurs systèmes de visualisation améliorée. Dans notre cas, les "perspectives wall" [22]et les arbres hyperboliques [16] peuvent apporter beaucoup à la clarté des informations de structure des documents MADEUS.

3.3 La vue d'exécution

La vue d'exécution est une vue particulière dans MADEUS. Elle possède un double rôle de visualisation : on peut visualiser des informations statiques ou dynamiques.

3.3.1 Informations dynamiques

L'information dynamique perçue dans la vue d'exécution est comme le nom de celle-ci l'indique la visualisation du déroulement du scénario temporel. C'est dans cette vue que l'on peut voir comment se déroule la présentation. Pour affiner la visualisation, la vue est munie d'une barre d'outil qui permet de voyager temporellement dans la présentation. Les principales fonctions de cette barrre sont l'exécution, l'arrêt, la pause d'un document. Elles sont complétées par des fonctions d'avance et de retour rapide permettant un contrôle des modifications visuelles apportées à une ou l'autre partie du document d'une part, et d'effectuer des sauts temporels dans la présentation lorsque celle-ci est jouée d'autre part. La barre d'outil n'est pas collée à la vue. Sinon, on perdrait une partie de l'information visible à l'endroit où elle est présentée.

3.3.2 Informations statiques

Pour ne pas augmenter inutilement le nombre de vues de l'éditeur, on peut aussi voir le positionnement spatial des différents objets apparaissants à l'écran. Il faut bien entendu arrêter le déroulement de la présentation pour effectuer des

modifications de placements d'objets. Il est de toute façon très difficile pour une personne normale de faire deux choses différentes en même temps, comme d'une part récuperer de l'information d'une présentation animée et encoder ces modifications d'autre part. On peut seulement voir leur placement par rapport à l'origine. Sans interaction de la part de l'utilisateur, on ne peut pas voir directement dans cette vue les contraintes spatiales reliant différents objets. On ne peut aussi voir que l'agencement spatial à un seul moment de la présentation à la fois. Pour voir un autre moment, il faut faire défiler la présentation et c'est seulement lorsque l'on se trouve en mode pause que l'on peut analyser à sa guise les informations spatiales des objets présentés lors de l'instant d'arrêt. Pour améliorer et mieux se rendre compte du placement spatial on pourrait pour un objet sélectionné donner les coordonnées spatiales de celui-ci. Ce qui est aussi fort utile, c'est de pouvoir ouvrir une table des contraintes synchronisées au moment de la présentation impliquant seulement les acteurs en présence. Ceci facilite franchement la visualisation du jeu de contraintes sans modifier le document. En effet lors de la sélection d'un objet dans la vue d'exécution, celui-ci se retrouve entouré par un rectange rouge. Les objets partageant une contrainte spatiale avec celui-ci sont quant-à eux entourés de vert. De ce fait en combinant les informations en présence, nous pouvons évaluer facilement les répercutions apportées par une modifcation éventuelle. Pour de plus amples renseignements sur le sujet, il est intéressant de consulter [4].

3.4 La vue Time Line

Nous allons étudier la vue Time Line d'un peu plus près que les autres.

3.4.1 Généralités

Affichage des médias

Les différents médias dans la vue Time Line sont représentés par des rectangles. Ces rectangles ont une hauteur prédéfinie quel que soit leur type. Ils ont une longueur proportionelle à leur durée préférée de présentation. Pour les différencier, nous avons choisi d'utiliser les conventions de représentation présentées à la figure 3.1:

Affichage des délais

Comme nous l'avons dit auparavant (Voir Chapitre document), certaines contraintes utilisent des délais. Par défaut, ceux-ci sont de durée variable (c'est le formateur temporel qui les fixe). Le créateur du document peut aussi les fixer Voir figure relation Allen et autre. La figure 3.2 montre les représentations utilisées.

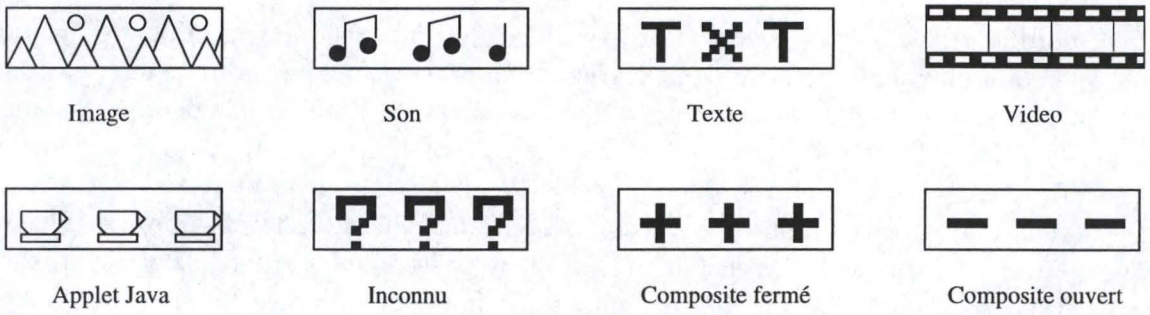

FIG. 3.1 – *Conventions de représentation des médias*

FIG. 3.2 – *Conventions de représentation des delais*

délai variable : Nous avons choisi une métaphore de ressort pour bien montrer que le délai n'est pas fixe.

La sémantique de l'affichage d'un délai variable peut être interprétée comme la suivante : Nous avons devant les yeux un scénario temporel possible et correct. Mais, si les objets accolés aux délais variables possédaient un autre instant de début de présentation (1 sec plus tôt, une plus tard), cette configuration resterait cohérente avec les contraintes données par l'auteur. L'exécution reste aussi parfaitement valide. Un petit exemple est présenté à la figure 3.3

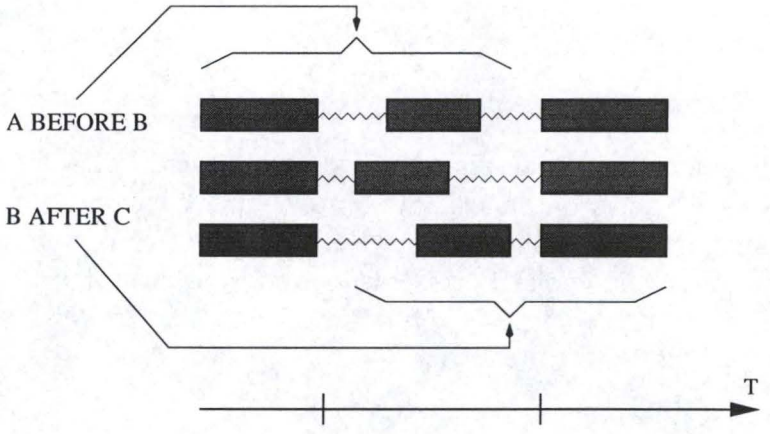

FIG. 3.3 – *Visualisation particulière d'une infinité de solutions*

3.4.2 Affichage des contraintes

Les contraintes étant l'élément primordial du scénario temporel, il est très important de les afficher.

Contraintes d'Allen

Pour ce faire, nous avons choisi de les représenter à l'écran par des lignes verticales. Une Contrainte temporelle peut être composée plusieurs lignes verticales. A chaque noeud du graphe ayant plusieurs arcs entrants ou sortants, on va relier les extrémités des rectangles ou bien les délais matérialisants les arcs, par une ligne verticale. Suivant le placements des différents objets à l' écran, il parait utile d'utiliser cette méthode comme on peut le voir sur la figure 3.4.

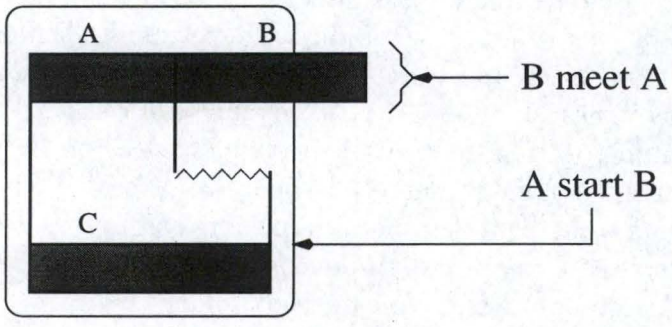

FIG. 3.4 – *Visualisation des contraintes d'Allen*

Contraintes supplémantaires

Nous avons choisi une différenciation pour représenter les contraintes parmin, parmax et parmaster. Ces contraintes sont aussi représentées par des lignes verticales, mais sur une de celles-ci, nous avons ajouté une flèche pour bien montrer que la fin d'un élément entraîne la fin de l'autre. La flèche étant dirigée vers l'élément coupé, si celui-ci est identifiable. Dans le cas contraire, l'utilisation d'une double flèche semble appropriée.

Remarques

En parcourant les conventions de représentations, on voit qu'il est difficile de visualiser une contrainte en entier. Pour avoir plus facile, il existe dans l'éditeur MADEUS une fenêtre dans laquelle sont répertoriées les relations existantes entre les différents objets du document.

Il pourrait se poser un problème de coupures de médias qui n'ont rien à voir avec la contrainte représentée, mais ce problème sera traité ultérieurement dans le chapitre 6.

3.4.3 Affichage du document

Ici nous allons voir les mécanismes qui se cachent pour afficher le document dans la vue Time Line. Cette partie est très importante, cest pouquoi nous allons lui consacrer un chapitre entier (Voir chapitre 6)

3.4.4 Interactions de l'utilisateur

L'utilisateur, par son action, peut visualiser des informations non disponibles au simple regard de la vue.

Ouverture/ fermeture de composite

En cliquant sur un composite, celui-ci s'ouvre ou se ferme suivant le cas. La méthode de visualisation adoptée a été mise en place pour bien se retrouver dans l'affichage. Nous sommes partis du principe que l'affichage était divisé en lignes. Si on ouvre un composite à la ligne X, alors, toutes les informations se trouvant sur les lignes plus basses (fait référence à un affichage de haut en bas et non à une numérotation de celles-ci) seront décalées de la hauteur prise par le composite ouvert comme présenté à la figure 3.5 2 et 3. Pour la fermeture, on applique le même principe à l'envers. Si plusieurs composites sont ouverts sur la même ligne,lors de la fermeture d'un d'entre eux, on remonte le plus possible les rectangles deplacés auparavant. Voir figure 3.5 3 et 4.

Informations de déplacements

MADEUS travaille avec des contraintes. Le jeu de contraintes étant important, il se peut que nous ne voyons pas bien les degrés de liberté de certains médias. Pour ce faire, en cliquant sur l'un d'entre eux, nous pouvons voir ceux qui y sont solidaires et l'intervalle de déplacement possible pour cet objet. Pour de plus amples renseignements : [31]

Visualisation des contraintes

Si dans la boîte de dialogues contenant les différentes contraintes temporelles exprimées sous formes textuelles, on sélectionne l'une d'entre elle, alors les différents médias, les délais impliqués et les lignes verticales utilisées à la liaisons des éléments cités précédemment vont changer de couleur jusqu'à ce que la sélection soit terminée.

3.5 Synchronisation des vues

MADEUS étant un editeur multi-vues, il permet d'en synchroniser certaines entre elles. Si elle le sont, une modification dans l'une d'elles va dans un premier

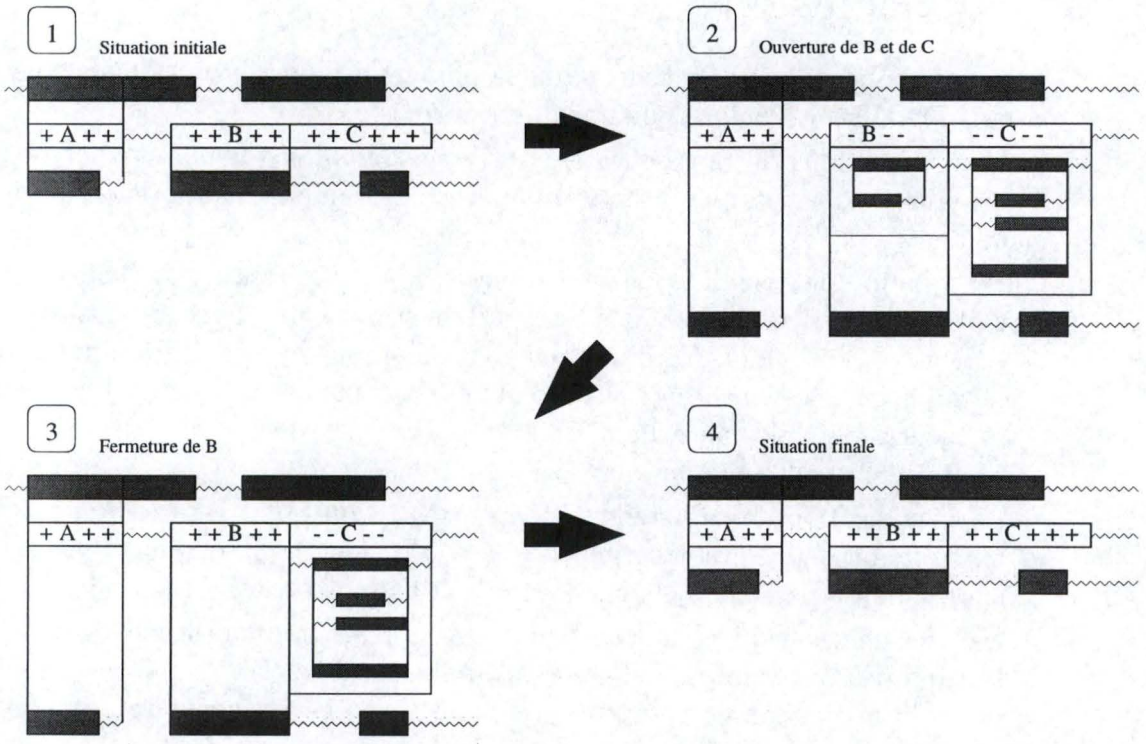

FIG. 3.5 – *Modification de l'affichage suite à une ouverture/fermeture d'un composite*

temps, si nécessaire, (information pertinente et nécessaire au bon fonctionnement des autres vues) modifier le document de référence de l'application. Dans un second temps, les modifications apportées au document vont être envoyées suivant le type d'informations visé à un manager spécialisé qui, lui, va dispatcher les modifications aux documents étendus des autres vues synchroniéees et à celui de la vue responsable de la modification. Comme cela, en procédant en deux temps, on est sûr de garantir une cohérence des données entre les différents documents.

3.6 Désynchronisation des vues

A chaque fois que l'on ouvre une vue dans l'éditeur, par défaut, celle-ci est synchronisée aux autres vues ouvertes. Il est possible de désynchroniser une vue par rapport à une/aux autre(s). Ce procédé est très pratique. Il peut permettre d'effectuer des modifications dans une vue sans que celles-ci soient prises en compte par le reste de MADEUS. Comme cela, si un test de modification du scénario temporel procure un ordonnancement non désiré, seul le document étendu de la vue sera modifié. De ce fait, le document de référence contiendra toujours la bonne version.

3.7 Conclusion

Dans ce chapitre, nous avons examiné la plupart des possibilités de visualisation dans MADEUS. Voici ce que nous pouvons en retirer :

- Les conventions de représentations utilisées dans les différentes vues pour visualiser les objets permettent un aperçu rapide de la composition du document.
- Le principe de visualisation multi-vues est très intéressant. En effet, cela évite à l'utilisateur un surcharge cognitive si plusieurs de ces informations étaient intégrées au seing d'une seule vue. Comme on a pu l'examiner dans le chapitre précédent, certains éditeurs n'intègrant pas ces facilités donnent beaucoup plus de mal à l'auteur lors de la réalisation d'une présentation pour qu'elle soit à sa convenance.
- La visualisation des différentes informations d'un objet est facilitée grâce au principe de syncrhonisation des vues. En effet, l'auteur peut d'un seul coup d'oeil repérer l'objet traité dans les différentes vues de l'application.
- La vue d'exécution et la vue Time Line intègrent un procédé de visualisation des contraintes. Cet outil est indispensable. Sans celui-ci, l'auteur pourrait ne pas connaître l'existence d'une ou plusieures contraintes ou bien en confondre certaines insérées dans le scénario temporel.
- Un problème déjà rencontré dans la plupart des autres systèmes auteurs reste à l'affiche ; la gestion de la mise à l'échelle. Sans un moyen performant pour combler cette lacune, un éditeur ne peut être efficace qu'avec de petits documents. MADEUS propose un début de solution en permettant d'ouvrir et de fermer certains composites à l'écran dans les fenêtres dédiées aux dimentions structurales et temporelles. MAlheureusement, ceci n'est pas suffisant.

Si l'on procure à l'auteur une visualisation adéquate des informations pertinentes du document, les processus d'édition devrait être intuitifs tout en permettant un large éventail de possibilités. Nous allons voir dans le chapitre suivant si MADEUS possède les capacités nécessaires pour satisfaire à cette règle.

Chapitre 4

Edition dans MADEUS

4.1 Introduction

MADEUS est un éditeur de documents multi-vues. Le chapitre précedent présentait chacune des vues constituant l'éditeur. Maintenant, nous allons nous intéresser aux capacités d'éditions proposées par MADEUS dans chacune d'elles.

L'éditeur se base sur un concept fondamental : Une édition particulière est favorisée pour un type de vue donnée. La répartition est la suivante :

- Edition spatiale : vue d'exécution
- Edition de la structure : vue hiérarchique
- Edition temporelle : vue Time Line
- Edition des hyperliens : vue Time Line

Il est aussi à noter qu'une édition particulière peut également dans certains cas être effectuée dans d'autres vues. Ce concept est à envisager si une opération d'édition semble plus intuitive dans une autre vue que la vue de référence initiale.

Dans ce chapitre, nous allons nous concentrer essentiellement comme dans le reste du travail sur la dimension temporelle de l'édition. Toutefois, lorsque la structure interne du document contenant le graphe temporel subit certaines modifications, celles-ci vont être explicitées un peu plus en détail.

4.2 Edition de la structure

Pour éditer la structure du document, la vue hiérarchique semble la plus indiquée. Dans cette vue, on peut facilement ajouter, retirer ou bien déplacer un objet. Comme nous travaillons sur la structure, il ne faut pas oublier de mettre à jour le document interne de la vue. Par le système de synchronisation des vues, les différentes informations modifiées vont êtres transmises aux autres vues sans rien faire. Nous allons voir ce qu'il faut modifier au document avec les différentes opérations d'édition

4.2.1 Ajout d'un objet

- Il faut créer un document pour l'objet lui même.
- Il faut le rattacher au composite englobant en mettant à jour son sous-graphe et le lien vers son père.
- Il faut ajouter un lien dans la liste des fils du composite englobant cet objet
- Il faut mettre à jour le graphe du composite englobant

4.2.2 Suppression d'un objet

Avant d'éliminer cet objet, il faut faire attention. MADEUS étant un éditeur à base de contraintes, il faut s'assurer que l'objet n'est pas impliqué dans une ou plusieurs contrainte(s) spatiale(s) ou temporelle(s). Si c'est le cas, il faut d'abord éliminer ces contraintes (pour ce faire, il faut avoir une copie des listes de contraintes dans la vue hiérarchique)et puis seulement après, nous pouvons supprimer l'objet proprement dit. Il ne faut pas oublier de

- supprimer le lien dans le document père.
- mettre à jour le graphe du document père.
- supprimer l'objet proprement dit.

Il paraît aussi évident que l'on ne peut supprimer un composite que s'il est vide. Dans le cas contraire, on peut envisager de supprimer tous les objets internes à celui-ci.

4.2.3 Déplacement d'un objet

La procédure est plus ou moins la même que pour la suppression. Il faut éliminer les contraintes dans lesquelles l'objet est impliqué. Puis il faut effectuer les opérations suivantes sur le document interne :

- Il faut le rattacher à son nouveau document père.
- Il faut supprimer le lien dans la liste des fils de son ancien document père
- Il faut mettre à jour les graphes des 2 documents pères
- Il faut mettre à jour son sous-graphe

4.2.4 Remarque

Les manoeuvres d'édition de la structure peuvent aussi s'effectuées dans les autres vues. En effet, lors d'une modfication du scénario, l'auteur trouvera plus intuitif d'insérer un objet dans la vue de travail actuelle. On pourrait donc envisager la chose dans la vue Time Line et la vue d'exécution. L'insertion d'un objet dans la vue Time Line le placera dans le composite principal si aucun autre n'a été sélectionné. Par contre la décision de placement dans la vue d'exécution sera reportée lors de l'assignation de la première contrainte pour cet objet.

4.3 Edition spatiale

La vue d'exécution permet de faire facilement des modifications spatiales. On peut voir 2 sortes d'édition spatiale dans MADEUS

4.3.1 Changement de la position d'un objet

Dans MADEUS cette opération se fait par un "glisser déplacer" de l'objet sur la surface de la vue d'exécution. Il faut aussi faire attention, car si il existe une contrainte spatiale dans laquelle il est impliqué, le ou les objets liés vont tous bouger avec le déplacement du premier.

4.3.2 Edition des contraintes spatiales

Ajout et suppression de contraintes directement dans la vue. Il suffit alors de sélectionner deux objets dans la vue et d'apporter les modifications voulues. Pour ce faire, le plus facile serait d'avoir pour la liste d'objets sélectionnés une liste des relations correspondantes. De là, on peut supprimer facilement une contrainte. On peut aussi éviter d'avoir 2 fois la même contrainte entre 2 objets.

4.4 Edition temporelle dans la vue Time Line

Le but de l'édition temporelle est de pouvoir modifier facilement les propriétés impliquant la dimension temporelle dans un document multimédia. Dans cette section, nous nous concentrerons sur la modification de la durée d'un objet à présenter. Il manquera à cette section une partie importante, celle de l'édition des contraintes temporelles dans MADEUS. Ce sujet étant assez complexe et l'éditeur étant encore en développement, nous parlerons brièvement de ces fonctionnalités dans le chapitre suivant énumérant quelques perspectives pour l'avenir.

Dans cette section, nous allons donc voir une manière de procéder pour arriver à modifier la durée d'un objet. Dans la suite de l'exposé, pour parler de la modification d'une durée, nous utiliserons communément le terme de retaillage. Ce terme est employé puisque nous travaillons sur des composants visuels et puisque l'on peut très bien imaginer cette action pour un média. Voici les différentes étapes

4.4.1 Remarques préliminaires

La durée d'un objet n'est pas une valeur fixe; elle est exprimée sous la forme d'un intervalle comprennant en plus une valeur préférée. Dans ce chapitre, nous allons raisonner sur la valeur préférée de présentation d'un objet et non sur l'intervalle entier. Ce choix a été influencé d'une part par la représentation de l'objet dans la vue Time Line et d'autre part par le fonctionnement de la machine

d'exécution. En effet, Dans la vue Time Line, la longueur des objets à l'écran est proportionnelle à la valeur de présentation préférée. Pour garder une cohérence dans l'affichage et ne pas embrouiller les esprits de l'auteur, il est plus facile de modifier les valeurs préférées. De plus, la machine d'exécution, lors de la présentation du document essaie au maximun de respecter les valeurs préférées assignées aux différents médias. Ceci n'est pas garanti tout le temps à cause de la présence d'objets incontrôlables dans le document.

Le retaillage d'un objet incontrôlable signifierait soit un changement dans le rythme de présentation soit une coupure nette de la présentation avant sa fin ou bien une boucle de présentation du média qui dans sa Xème visualisation sera tronquée au temps voulu.

4.4.2 Selection du média

Dans la vue TimeLine, pour sélectionner un objet à retailler, il suffit de se placer au dessus de celui-ci. Si le pointeur de la souris est sur une zone partant d'une extremité du rectangle représentant le média et ayant une longeur de +- 5% de sa taille totale, alors, l'apparence du curseur changera comme on peut le voir à la figure 4.1. A partir de ce moment, le retaillage de l'objet est possible. Pour l'effectuer, il faut faire un "drag" avec la souris jusqu'à ce que la nouvelle dimension convienne à l'auteur.

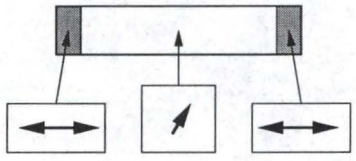

FIG. 4.1 – *différentes formes du curseur*

4.4.3 détérmination de la structure solidaire de l'objet.

Dans cette étape, nous allons simplement définir la structure solidaire de l'objet à retailler. Par structure solidaire, nous entendons les différents éléments (aussi bien les médias que les délais) qui bougeraient avec lui si on déplaçait celui-ci sur la ligne du temps. C'est-à-dire tous les médias et les délais impliqués dans un jeu de contraintes obligeant cette structure inséquable. Pour ce faire, nous allons utiliser le même procédé que celui utilisé dans la sous-section 3.4.4. A la fin de cette étape, nous allons nous retrouver avec deux listes comprenant respectivement les délais permettants d'amortir l'instant de début des objets et les délais permettants d'amortir l'instant de fin de leur présentation. Pour plus de facilité dans la suite de l'exposé, nous allons renommer ces deux listes par les mentions suivantes: liste des délais avançants et liste des délais retardants.

Nous avons choisi ces deux termes en nous basant sur la sémantique apportée à la visualisation d'un délai dans la vue. Lorsque l'on regarde la vue Time Line, si on déplaçait la structure solidaire, la taille des délais entourant celle-ci changerait. Si cet ensemble d'éléments est déplacé vers la gauche, alors l'utilisateur peut avoir comme première impression que le moment de présentation va être avancé (d'ou le nom de délais avancants). Il en est de même pour les délais retardants. Nous allons encore employer une autre convention de nommage. Tous les délais présents dans la structure solidaire seront appelés des délais coincés parce qu'il se comportent comme des délais fixes. Leur perte de flexibilité est due au jeu de contraintes.

4.4.4 Calcul de l'intervalle de retaillage possible

Comme nous sommes dans une vue visuelle, pour retailler un objet nous allons permettre à l'utilisateur de le faire de la manière la plus intuitive possible. Généralement, il paraît assez intuitif de vouloir allonger ou diminuer un objet placé à côté d'un délai. Pour le confort de l'utilisateur, nous allons donc permettre un retaillage vers la droite ou vers la gauche. Pour montrer à l'auteur la marge dont il dispose, il faut afficher à l'ecran l'intervalle de retaillage possible pour l'objet.

Rappelons encore une fois que MADEUS est un éditeur basé sur des contraintes temporelles. Le jeu de contraintes peut dans certains cas aussi limiter l'intervalle de retaillage d'un objet. Cet intervalle pourrait même être nul. Il faut donc être très vigilant lors du calcul de celui-ci. Si il existe un chemin ne passant pas par l'objet à retailler reliant deux autres médias, l'un accesible par la borne d'entrée et l'autre par la borne de sortie de cet objet, sans que le chemin contienne de délais alors, dans ce cas, son intervalle de retaillage est nul. Dans le cas contraire, l'intervalle de retaillage peut être divisé en deux parties. La première servant pour l'augmentation potentielle de la durée de l'objet et la seconde pour sa diminution. Le but de la suite de ce travail sera donc de déterminer séparement les deux demi-intervalle et de les juxtaposer pour connaître l'intervalles total. Pour ce faire, nous devons récolter les informations utiles.

détérmination des délais utiles à la construction de l'intervalle

Nous avons déjà analysé une partie des délais mis à notre disposition précédemment. Ici, nous allons affiner la classification des délais coincés. On peut dans cet ensemble de délais identifier cinq nouvelles catégories :

- délai d'extension_droite : les délais du sous-graphe limitant l'augmentation de la durée du média pour un retaillage à droite.
- délai d'extension_gauche : les délais du sous-graphe limitant l'augmentation de la durée du média pour un retaillage à gauche.

- délai de diminution_droite : les délais du sous-graphe limitant la diminution de la durée du média pour un retaillage à droite.
- délai de diminution_gauche : les délais du sous-graphe limitant la diminution de la durée du média pour un retaillage à gauche.
- délai neutre : les délais du sous graphe n'intervenant pas dans ces calculs et les délais éliminés des catégories précédentes.

Pour un retaillage à gauche ou à droite, entre les catégories d'extension et de diminution, la cardinalité de leur intersection est nulle.

Entre la catégorie neutre et les autres, la cardinalité de leur intersection est aussi nulle. Pour les autres cas de figure, les délais en commun à plusieurs catégories sont possibles.

Une fois les différents éléments d'analyse expliqué, il ne reste plus qu'à déduire les informations pertinentes pour les différents cas pouvant se présenter.

Retaillage à droite

Composantes possibles pour l'intervalle diminuant la durée du média

- nulle
- délai de diminution_droite
- valeur préférée- borne inférieure (limitation de l'intervalle de durée)

Composantes possibles pour l'intervalle augmentant la durée du média

- nulle
- délai retardants de la structure solidaire
- délai d'augmentation_droite
- borne supérieure - valeur préférée (limitation de l'intervalle de durée)

.

Retaillage à gauche

Composantes possibles pour l'intervalle diminuant la durée du média

- nulle.
- délai de diminution_gauche
- valeur préférée- borne inférieure (limitation de l'intervalle de durée)

Composantes possibles pour l'intervalle augmentant la durée du média

- nulle.
- délai avançants de la structure solidaire
- délai d'augmentation_gauche
- borne supérieure - valeur préférée (limitation de l'intervalle de durée)

.

Nulle part, nous ne parlons des délais neutres. Or, de par leurs descriptions, on voit qu'ils contiennent aussi des délais éliminés des autres catégories. Voici ci-dessous des critères d'élimintation :

La sémantique d'un délai : Un délai ne peut jamais être nul. De ce fait les délais ayant une longueur égale à une unité de temps ne pourront pas entrer

	Point de départ	point d'arrivée	Chemin OK	Chemin KO
retaillage à droite				
diminution	BS média	BS délai	D et 3	D et 4
augmentation	BS média	BI délai	A et 1	A et 2
retaillage à gauche				
diminution	BI média	BI délai	B et 1	B et 2
diminution	BI média	BI délai	B et 1	B et 2

TAB. 4.1 – *vérification de l'utilité des délais*

dans les calculs. Pour effectuer le choix sans problème, il faut réduire tous les délais d'une unité temporelle avant de commencer la classification. Dans ce cas, on éliminerait les délais nuls.

fausses informations : Dans ce cas, certains délais choisis n'influencent pas la décision finale. On rencontre le cas lorsque sur un chemin rejoignant une des extremités de l'objet à retailler au délai considéré, ce chemin passe par un autre délai, celui-ci joue le rôle d'un agent absorbant. Ceci implique que le délai n'est pas à considérer dans le calcul du demi-intervalle. Pour que le chemin soit valide, il ne peut pas emprunter un des deux objets. Il ne peut comprendre que l'une de leurs bornes. Le tableau 4.1 répertorie les différents chemins à vérifier. on peut voir une illustration à la figure 4.2

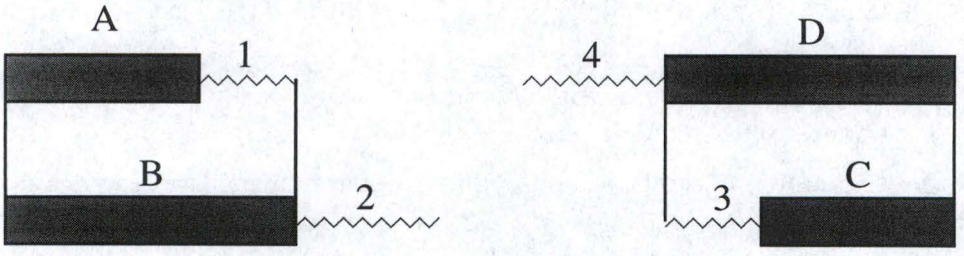

FIG. 4.2 – *vérification de chemin*

Pour connaître la valeur définitive des demi-intervalles, nous allons prendre le minimun des différentes composantes.
Logiquement, si on reste dans l'intervalle de durée spécifé, alors la cohérence temporelle doit être respectée. Si on voulait sortir des bornes, dans ce cas, il faut relancer un test de cohérence temporelle.

Remplissage des différentes listes

Il ne nous reste qu'à étudier comment déterminer les éléments de chaque listes. Pour ce qui est de la liste des délais avancants et des délais retardants, elles ont été définies dans la sous-section 4.4.3 à la page 44. Nous allons construire les

autres ci-dessous. En pratique, nous allons utiliser une variante de l'algorithme développé par Laurent Tardif [31] déjà utilisé précédemment. Dans notre cas si on l'applique à la lettre, on obtiendra les mêmes résultats que pour la structure solidaire. Volontairement, nous avons retiré une possibilité de passage pour le parcours d'arbre. On retirera donc le média à retailler lui-même. Il faudra lancer deux fois l'algorithme. La première exécution commencera sur un arc dont une des extremités est adjascente au noeud de départ de l'objet à retailler. Et la seconde aura comme point de départ un autre arc dont une des extrémités est commune au noeud de fin de l'objet à retailler. Cette double application va donner comme résultat six liste de délai. Voici leur classification :

- Liste des délais à droite associée au noeud d'entrée (liste 1)
- Liste des délais à gauche associée au noeud d'entrée (liste 2)
- Liste des délais entourés associée au noeud d'entrée (liste 3)
- Liste des délais à droite associée au noeud de sortie (liste 4)
- Liste des délais à gauche associée au noeud de sortie (liste 5)
- Liste des délais entourés associée au noeud de sortie (liste 6)

Nous entendons par liste de délais à gauche, à droite et entourés les définitions suivantes :

- Un délai à gauche est un délai atteint par sa borne inférieure lors du parcours du sous arbre
- Un délai à droite est un délai atteint par sa borne supérieure lors du parcours du sous arbre
- Un délai entouré est un délai atteint par sa borne inférieure et par sa borne supérieure dans le même parcours.

Une fois ces données récoltées, nous pouvons commencer l'analyse des éléments de ces listes.

Détermination d'un intervalle de déplacement nul

Cette double opération permet de découvrir si le retaillage est possible. Si on retrouve dans les deux listes des délais entourés des éléments en commun, alors l'intervalle de retaillage est nul. Dans ce cas, on remarque bien qu'entre les délais approchés par l'extremité inférieure ou supérieure de l'objet à retailler, on passe par des délais communs. Une seule explication est possible. Il existe bien un chemin ne passant pas par l'objet à retailler reliant des arcs des deux "hémisphères". Comme mentionné dans la sous-section 4.4.4 à la page 45, nous avons à faire à un intervalle nul. On peut aussi trouver un intervalle nul si :

- un délai est présent dans la liste 1 et dans la liste 4 (figure 4.3 B et D)
- un délai est présent dans la liste 2 et dans la liste 5 (figure 4.3 A)

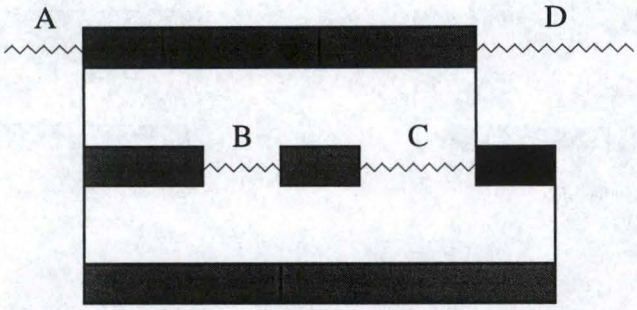

FIG. 4.3 – *délais empéchants le retaillage*

Détermination des délais d'augmentation

Retaillage à droite
Un délai est un délai d'augmentation_droite si :

- il n'est présent que dans la liste 1
- il n'est présent que dans la liste 4
- il est présent dans la liste 1 et dans la liste 5 (figure 4.4)

Retaillage à gauche
Un délai est un délai d'augmentation_gauche si :

- il n'est présent que dans la liste 2
- il n'est présent que dans la liste 5
- il est présent dans la liste 2 et dans la liste 4 figure 4.5

Détermination des délais de diminution

Retaillage à droite

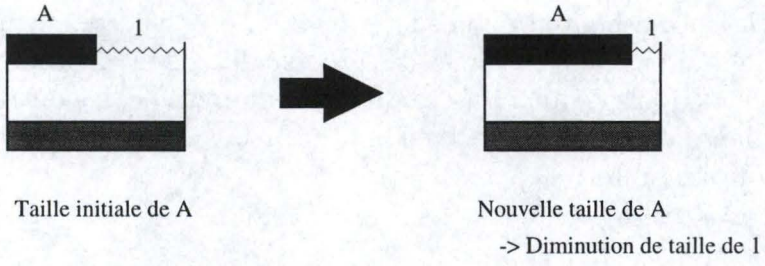

FIG. 4.4 – *délais à droite limitant l'augmentation de la taille*

Un délai est un délai de diminution_droite si :

- il n'est présent que dans la liste 2
- il n'est présent que dans la liste 5

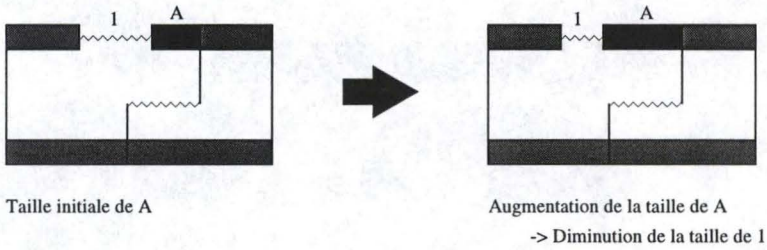

FIG. 4.5 – *délais à gauche limitant l'augmentation de la taille*

- il est présent dans la liste 2 et dans la liste 4

Retaillage à gauche
Un délai est un délai diminution_gauche si :

- il n'est présent que dans la liste 1
- il n'est présent que dans la liste 4
- il est présent dans la liste 1 et dans la liste 5

Si on remarque bien, le cas d'un retaillage à gauche est l'inverse de celui d'un retaillage à droite. C'est à dire les délais à gauche pour un sont les délais à droite pour l'autre et vice et versa. Donc pour la détérmination de l'intervalle total de retaillage, si celui-ci n'est pas nul, ce sont les autres composantes qui vont être détérminantes pour sa limitation. Sinon, on aurait pu employer la même méthode pour les deux.

4.4.5 Edition proprement dite

Cette étape est la plus simple de toutes. C'est à ce moment que l'auteur change la durée préférée de l'objet à retailler. Si, dans son action, celui-ci dépasse l'intervalle détérminé au préalable lors du glisser déplacer de la souris, la modificaton s'arrêtera bien évidemment à la valeur minimale ou maximale de l'intervalle délimité. Une fois le bouton de la souris relaché, on peut commencer à modifier les valeurs de présentation préférées des objets allongés ou bien diminués. Il est à remarquer que ces objets modifiés seront le média retaillé, les délais énoncés suivant le cas dans la sous-section 4.4.4 à la page 45 et suivantes. Voici les différentes configurations possibles :

retaillage à droite

- extension du média
  - délai à gauche ne bouge pas
  - délai à droite diminue
- réfraction du média
  - délai à gauche diminue
  - délai à droite ne bouge pas

retaillage à gauche

- extension du média
  - délai à gauche diminue
  - délai à droite ne bouge pas
- réfraction du média
  - délai à gauche ne bouge pas
  - délai à droite diminue

Une fois ces objets mis à jour, si on rafraîchit l'affichage, on va remarquer des incohérences de placements, des chevauchements de médias et de délais. Il faut donc mettre toutes les positions spatiales (coord en X) des objets ayant leur instant de présentation préféré modifé à cause du retaillage à jour.

4.4.6 Mise à jour des coordonnées d'affichage

Pour ce faire, il faut tout d'abord identifier des "piliers". C'est-à-dire des éléments du grahe ne ressentant pas les modifications à apporter. Potentiellement, tous les arcs compris dans la structure solidaire de l'objet à retailler peuvent changer de positions. Ceci veut dire que l'on peut se baser soit sur l' instant de début des délais avançants ou bien l'instant de fin des délais retardants. Les différents cas et les points de départ de la mise à jour sont présentés dans le tableau 4.2.
Si un de ces délais faisait partie des objets retaillés, ce n'est pas grave. On connait sa nouvelle taille et un de ses noeuds d'extrémité fixe. De ces deux informations on peut déduire les coordonnées à assigner au document relié à l'arc(abcisse du document suivant- taille du délai retaillé). Pour la suite, il suffit de parcourir tous les arcs de la structure et de détérminer leurs nouvelles positions spatiales en utilisant le même petit calcul. Les valeurs répercutées partant sur des bases cohérentes, il ne doit pas y avoir de problèmes comme par exemple une abcisse différente pour le placement d'un même noeud. Les extremités du parcours sont présentées suivant les cas dans le tableau 4.2

	point de départ du parcours	point d'arrêt du parcours
retaillage à droite		
extension du média	délais retardants	délais avançants
refraction du média	délais retardants	délais avançants
retaillage à gauche		
extension du média	délais avançants	délai retardants
refraction du média	délais avançants	délai retardants

TAB. 4.2 – *sens du parcours de mise à jour*

4.5 Edition temporelle dans les autres vues

Rappelons tout d'abord qu'il n'est possible d'éditer les propriétés temporelles d'un média dans une vue donnée, que si le document étendu associé à celle-ci possède ces informations. Plus précisement, il doit tenir compte des intervalles de durée de présentation des objets.

Nous allons présenter ci-dessous les différentes étapes nécessaires pour éditer une durée temporelle dans une vue différente de la Time Line.

4.5.1 Préparation de l'édition

Cette première étape commence simplement par la sélection du média à retailler. Il faut faire bien attention lors de la sélection. L'édition peut être réalisée seulement si un seul objet est sélectionné. Si ce n'est pas le cas, il faudra auparavant désélectionner les "intrus". Ensuite, le seul moyen d'accéder aux informations temporelles est de faire apparaître à l'écran la fenêtre de propriétés d'un objet et de se placer sur l'onglet rassemblant les informations voulues. Dans la fenêtre de propriétés, nous allons ajouter un bouton supplémentaire pour démarrer le calcul de l'intervalle de retaillage. En effet, cette fenêtre peut être invoquée dans des situations ne comprenant pas une phase d'édition temporelle. Dans ce cas, effectuer les calculs de la détermination de l"intervalle de retaillage s'avère inutile et cela permet d'économiser les resources systèmes.

4.5.2 Détermination de l'intervalle de retaillage

Le calcul de l'intervalle de retaillage ne se passe pas tout à fait de la même manière que dans la vue Time Line. En effet, l'édition de propriétés ne permet pas une double édition (retaillage à gauche ou bien à droite). Pour la détermination de l'intervalle, nous allons quand même récupérer quelques informations de la même manière que celle vue à la section précédente. Une liste des informations communes aux deux méthodes est présentée ci-dessous :

- le minimun des délais d'augmentation_gauche et d'augmentation_droite
- le minimun des délais de diminution_gauche et de diminution_droite
- le minimun des délais avançants
- le minimun des délais retardants
- la valeur nulle pour l'intervalle si c'est le cas.
- Borne inférieure du média
- Borne supérieure du média

Avec ces différentes données, nous allons pouvoir calculer l'intervalle limitant le retaillage Si l'intervalle n'est pas nul, la détérmination des bornes se fera de la

manière suivante :

- La borne supérieure prendra la valeur suivante : min(délais augmentation_gauche, délais avancants) + min(délais augmentation_droite, délais retartdants).
- La borne inférieure recevra la valeur minimale des composants suivants :
  - le minimun des délais de diminution_gauche
  - le minimun des délais de diminution_droite

Les bornes comme exposées ci-dessus sont des bornes relatives. C'est-à-dire qu'elles sont exprimées par rapport à la valeur préferée de la durée de présentation du média. Si l'on veut connaître le vrai intervalle, il faut pour la borne supérieure lui ajouter la valeur de la duréé préférée et pour la borne inférieure, retirer sa valeur de la durée préférée. Evidemment, pour que l'intervalle de retaillage soit valide, il faut en plus que celui-ci soit inclus dans les bornes de durée de l'objet.

4.5.3 Affichage de l'intervalle

Pour afficher cet intervalle, nous allons présenter deux solutions. La première est la plus simple à mettre en oeuvre. Elle consiste à ajouter dans la fenêtre de préférence deux boîtes de labélisation, l'une contenant la borne inférieure de l'intervalle et l'autre la borne supérieure de celui-ci comme on peut le voir à la figure 4.6. le placement spatial de ces boîtes montre bien que ces bornes sont comprisent dans l'intervalle affiché au dessus. Avec cette solution, la modification de la valeur préférée se fait en changeant la durée inscrite dans la boîte d'édition contenant l'ancienne valeur. Cette solution possède un petit inconvénient. En regardant l'affichage proposé, il n'est pas facile de se rendre compte de l'étendue de l'intervalle de retaillage et de sa position par rapport à l'intervalle de la durée.

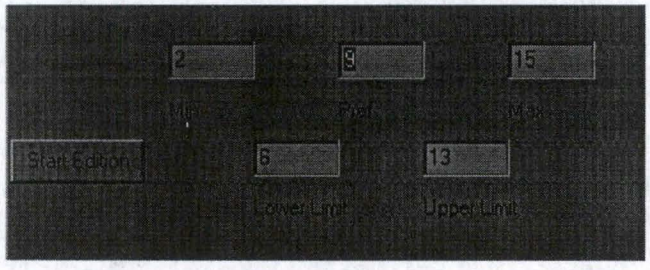

FIG. 4.6 – *Première possibilité d'affichage de l'intervalle*

La seconde solution va palier à cet inconvénient. Pour ce faire, nous allons ajouter à la fenêtre de préférence un espace graphique. Dans cet espace, va être dessiner l'intervalle de retaillage, mais proportionellement aux bornes de durée comme cela est montré à la figure 4.7. On va aussi ajouter en plus deux lignes verticales matérialisant les bornes du média, pour que l'auteur puisse bien se rendre

compte de l'espace de retaillage proposé par rapport à l'étendue de l'intervalle. Cette représentation est meilleure, mais il manque encore une chose : une troisième barre verticale proposant l'instant préferé de présentation du média. Comme cela on peut apprécier plus facilement l'influence de la modification par rapport à la valeur actuelle du média à retailler.

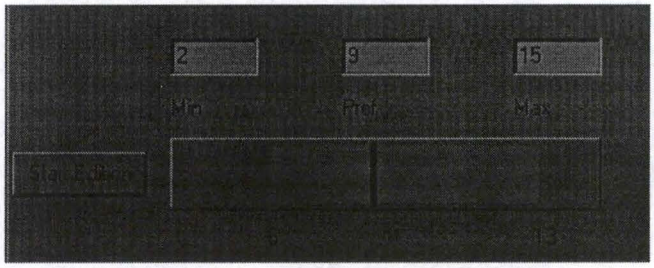

FIG. 4.7 – *Deuxième possibilité d'affichage de l'intervalle*

4.5.4 Edition proprement dite

L'édition se fera simplement en modifiant la valeur contenue dans la boîte d'édition incluant la valeur préférée de présentation du média à retailler. Il ne faut pas aussi oublier de faire un test pour voir si la nouvelle valeur est bien comprise dans les bornes calculées lors de la seconde étape.

Une solution beaucoup plus intuitive aurait pu être rétenue. Il s'agissait d'introduire un nouvel outil de modification des données : une jauge. L'édition aurait consisté à déplacer le curseur dont la position initiale montrerait la longueur actuelle du média. Un déplacement de celui-ci vers la gauche entraînerait une diminution de la durée tandis qu'un mouvement vers la droite produirait une augmentation de celle-ci. L'inconvénient majeur de cette solution, est que si on veut rester dans les bornes délimitées par le placement spatial des boîtes d'éditions de la durée (pour posséder les mêmes avantages de lisibilité et d'intuitivité procuré par la seconde solution présentée dans la section précedente) et si la taille de l'intervalle de retaillage est très grande, alors un cran de la jauge pourrait être supérieur à une unité de temps. Dans ce cas, l'affectation de certaines valeurs aurait été impossible. C'est pourquoi cette belle solution n'est pas à préconiser. Une illustration est présentée à la figure 4.8.

4.5.5 Modification de l'affichage

Les modifications s'effectuant dans une vue particulière, il faut laisser se dérouler le mécanisme de synchronisation pour que les changements soient répercutés dans la Time Line. A cet instant, la modification de l'information temporelle modifiée à été detectée dans la vue. Dans ce cas, il faut recommencer

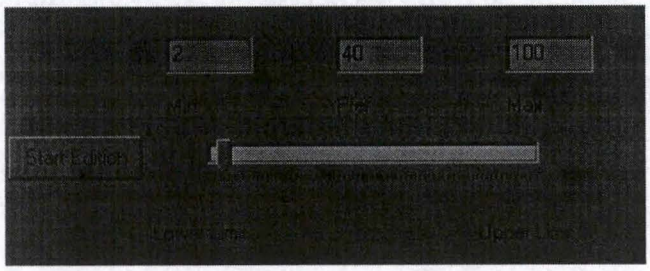

FIG. 4.8 – *Possibilité d'édition de l'intervalle non retenue*

une récolte des informations spatiales de placement comme exprimée à la section 6.2 à la page 68. Cette méthode etant beaucoup plus rapide qu'une approche de changements incrémentaux.

4.6 Remarque finale sur l'édition temporelle

Tout ceci pourrait aussi s'appliquer si le retaillage dépasse l'intervalle de présentation du média. Seulement, on devrait refaire les tests de cohérence temporelle pour valider l'inrtervalle trouvé. Si celui-ci n'est pas valide, il faudrait réessayer avec un plus petit jusqu'à ce que le test réussisse. Il se peut aussi qu'il faille modifier les intervalles des délais susceptibles d'être modifiés par le retaillage pour avoir la cohérence temporelle recherchée. Seulement à partir de cet instant, l'intervalle pourra être affiché à l'écran. La procedure se poursuivant normalement.

4.7 Conclusion

Dans ce chapitre nous avons énoncé une certaine partie des possibilités d'édition de MADEUS. Voici maintenant quelques commentaires caractésisant ces foctionnalités.

- Le choix de la vue dédiée pour l'édition de la structure parraît seulement pertinent lors de la phase de création du document. Pour une modification fine de la présentation, l'auteur préférera une autre vue attachée à la sémantique de la contrainte en cours de traitement.
- L'édition directe de la composante temporelle dans la vue Time Line permet d'offir à l'auteur un ensemble de solutions possibles matérialisées par un intervalle à l'affichage. En fixant la valeur préférée de l'objet dans cet intervalle, la cohérence temporelle du document est respectée. Si par contre on ne restraignait pas cet intervalle, l'auteur en effectuant une modification y perdrait sur deux tableaux. Tout d'abord on remarquerait une baisse des performances et une attente forcée à cause de la nouvelle vérification de

cohérence. Ensuite, on lui fait gagner du temps lors de la conception. Si le teste de cohérence échoue, il devrait respécifier la durée de l'objet à retailler et cela tant que les valeurs entrées ne conviennent pas.

Dans la suite de ce travail, nous allons nous intéresser à la visualisation des données temporelles. En effet, si l'affichage du scénario de la présentation n'est pas optimal, L'édition d'un objet ne peut se faire de manière aisée. Le graphe temporel est le support des informations à visualiser. C'est pourquoi nous allons travailler sur celui-ci et essayer de le représenter d'une façon idéale à l'écran. Pour ce faire, dans un premier temps nous passerons en revue l'état de l'art de la discipline au chapitre 5. Dans un second temps nous allons, vu les enseignements tirés de cette analyse, voir la solution retenue pour notre application. Afin de réaliser cette tâche, nous devons entre autre respecter certaines conventions afin d'obtenir un affichage optimal du scénario temporel. Le chapitre 6 relate ces propos.

Chapitre 5

Visualisation de graphes

5.1 Introdocution

Le but du "graph drawing" est double. D'une part, il faut assigner des coordonnées aux différents noeuds composants le graphe pour permettre leur affichage à l'écran. D'autre part, il faut placer les arcs d'une manière judicieuse pour procurer la meilleure visibilité de la structure possible. Comme nous pouvons le constater à la figure 5.1 un placement différent des noeuds dans le plan produit pour le dessin A une représentation agréable tandis que celle du dessin B est plus confuse. C'est pourquoi, dans ce chapitre, après avoir énoncé quelques concepts,

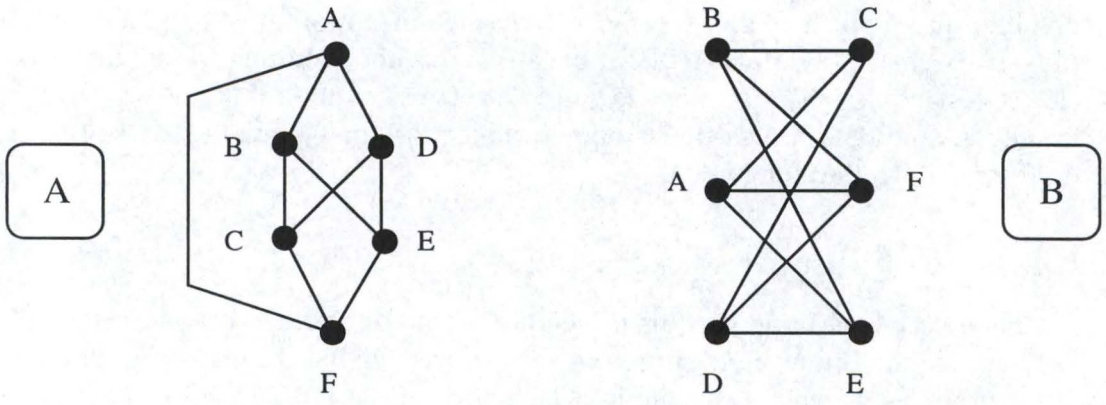

FIG. 5.1 – *Problème d'un bon placement de noeuds dans le plan*

nous allons faire un tour d'horizon des différentes méthodes de dessin de graphes. Une fois familiarisé avec celles-ci, nous en déduirons la plus prometteuse dans notre cas. Nous terminerons en l'analysant brièvement étape par étape.

5.2 Caractéristiques d'un graphe

Dans cette section, nous allons faire un bref rappel des caractérisques possibles d'un graphe. Cette caractérisation provient de [10] et de [12].

degré d'un graphe : C'est le nombre maximun d'arcs incidants à un noeud du graphe. Si celui-ci est de 4, on parlera aussi d'un 4-graphe.

digraph : C'est la contraction de "directed graph". Autrement dit un graphe orienté

graphe sans cycles simples : Un cycle est simple s'il n'utilise pas deux fois le même arc.

puit : C'est un noeud n'etant connecté à aucun arc sortant de lui. La notion de puit ne se rencontre que dans un "digraph"

st-digraph : C'est un "digraph" ayant une seule source et un seul puit dans son jeu de noeuds.

graphe connexe : Pour un graphe non orienté, on parle de connexité si il exsite pour n'importe quels noeuds du graphe un chemin les joignant deux à deux.

arbre : C'est un graphe connexe sans cycle

arborescence : c'est un arbre comportant un noeud spécial appelé la racine. Il existe entre celle-ci et n'importe quel autre noeud du graphe un chemin les joignants.

5.3 Dessin de graphe

Dans cette section, nous allons présenter différentes méthodes permettant de dessiner un graphe à l'écran. Voici ci dessous une liste reprennant les méthodes les plus utilisées dans la pratique. Pour ce faire, nous nous sommes basés sur la thèse [11] et les articles [30], [7]. Dans la suite de cette section, la documentation étant en anglais, nous avons gardé les appellations d'origines pour ne pas commettre de fautes de traduction.

5.3.1 Polyline drawing

Dans cette méthode de dessin, chaque arc composant le graphe est tracé à l'écran à l'aide d'une ligne simple ou bien d'une chaîne de segments de droite. Nous pouvons en voir un exemple a la figure 5.1 A. Pour la suite, nous allons définir le point de recontre de deux segments de droite appartenant au même arc comme un coude. Si une chaîne comprend trop de coudes, l'oeil humain rencontre des difficultées pour trouver les noeuds attachés à cet arc. Pour éviter cela, les chaînes dessinées doivent au plus comporter un maximun de 2 à 3 coudes. Les angles séparants les différentes droites matérialisant les arcs à l'ecran ne doivent pas êtres trop aigus, sinon la différentiation de ceux-ci est compromise. [10] et

[29] présente très bien ces aspects. Cette méthode est applicable pour tous les types de graphes cités précédement.

5.3.2 Straight-line drawing

Dans cette méthode, deux noeuds du graphe sont reliés par une droite rectiligne et continue [9] et [33]. On peut remarquer que c'est un cas particulier de "polyline drawing". Par ailleurs, cette méthode possède quelques petits désavantages. Si par exemple, il exite beaucoup de croisements à l'affichage, la lisiblilité de la structure est compromise, comme on a deja pu le constater à la figure 5.1 B. Dans d'autres cas, certaines informations se retrouvent cachées à l'affichage. En effet, si plusieurs arcs possèdent les mêmes noeuds d'extrémités, alors seulement un seul des differents arcs sera visible.

5.3.3 Orthogonal drawing

Dans cette methode, un arc est représenté par une chaîne de segments de droite. Les différents segments sont placés horizontalement ou bien verticalement à l'ecran. Aucune autre disposition n'est permise. On peut en voir une illustration a la figure 5.2 C et D. Une façon de créer ce genre de graphe est présentée dans [11]. Elle consiste à utiliser une technique de "visibility drawing" en premier lieu. Ensuite, il reste à modifier cette structure d'affichage pour arriver a une représneration du graphe ne comprenant que arcs composés de segments orthogonaux. Ces segments s'enchaînant avec le moins de coudes possible. Une illustration de ce procédé est presenté à la figure 5.2.

Evidemment, "l'orhtogonal drawing" n'est pas applicable pour des graphes ayant un degré supérieur à quatre.

Visibility drawing
Avec cette méthode on représente les noeuds par des lignes horizontales et les arcs par des trait verticaux joignants les différents noeuds extrémités de ceux-ci. On peut en voir un exemple à la figure 5.2 B.

5.3.4 Planar drawing

Cette méthode permet d'afficher un graphe sans croisements entre deux arcs. Généralement, pour ce faire, le graphe passe par une phase de "planarisation", ensuite il est affichable sans croisements comme nous pouvons le voir dans [23] et [32].

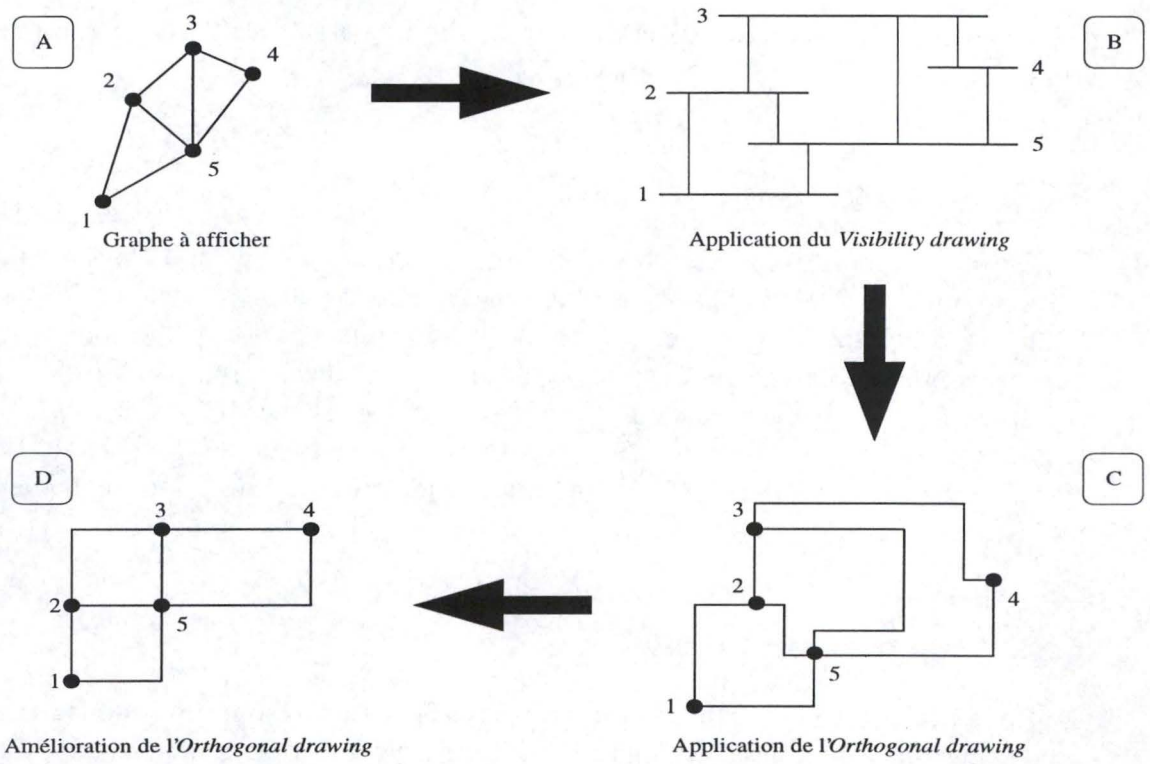

FIG. 5.2 – *Création d'une représentation d'Orthogonal drawing*

5.3.5 Hierarchical drawing

Dans un affichage hiéarchique, tous les noeuds appartenants à un même niveau sont placé sur une même ligne horizontale. Pour ce faire, il faut bien evidemment déterminer les différents niveaux. On peut procéder de la manière suivante :

- Si on a affaire à un "st-graph" ou bien à une arborescence, le niveau auquel appartient un noeud peut être déterminé en calculant sa distance par rapport à la racine du graphe comme présenté dans [12] et dans [28].
- Si on a affaire à un autre graphe orienté, il suffit d'identifier ses noeuds puits. Ceux-ci contsituent le dernier niveau de la hiérarchie. Pour déterminer les autres, nous allons retirer les arcs entrants aux noeuds puits et rechercher les nouveaux puits dans le sous graphe ainsi formé.

A chaque niveau, pour que la représentation reste lisible, il ne faut pas oublier de respecter une distance minimale entre deux noeuds adjascents.

5.3.6 Upwards drawing

Cette méthode ne s'applique qu'aux graphes orientés. Il existe quelques conventions de représentations.

- Les arcs sont représentés par des flèches.
- Le noeud de départ d'un arc sera place à l'écran en dessous de son noeud d'arrivée.
- l'orientation d'une flèche pointera toujours vers le haut.

A la figure 5.3, on peut voir un exemple et un contre-exemple. Une analyse concernant cette méthode peut être consulté dans [30].

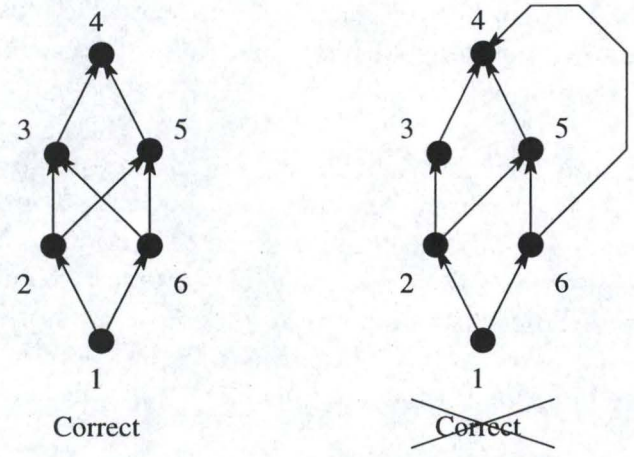

FIG. 5.3 – *Réprésentation d'un graphe en Upwards drawing*

5.3.7 Combinaison de méthodes

Cette liste non exhautive n'est pas la seule existante. On peut aussi affiner la représentation voulue. Pour ce faire, il est possible combiner les différentes méthodes présentées dans les sous-sections précédentes entre elles. Nous pourrions alors par exemple parler de

- Upwards planar Hierarchical Drawing
- Upwards Orthogonal Drawing
- ...

Pour de plus ample informations sur le sujet, [7] fourni une bibliographie complète sur les algorithmes implémantants les méthodes decrites ci-dessus.

5.4 MADEUS et le dessin de graphe

Le graphe temporel de MADEUS peut être caractérisé de la façon suivante :

- c'est un graphe orienté
- c'est un graphe sans circuits simples
- c'est un st-digraph
- c'est une arborescence

De plus, on peut ajouter que ses noeuds représentent les instants de début et de fin des médias et que les arcs fournissent leur durée.

Pour representer ce graphe temporel, il nous semble que notre analyse devrait se tourner vers la représentation d'un arborescence sous une forme hiérarchique. En effet, le graphe temporel est bien une arborescence. De plus, si on lui assigne à chaque noeud un instant de début relatif par rapport à la racine, nous pouvons alors le découper en plusieurs niveaux.

Plusieurs outils de visualisation de graphe comme par exemple AGD [23], DiVinci [6] et VGC [25] utilisent une méthode commune pour visualiser des arborescences hiérarchiques. Dans la suite de cette section, nous allons résumer la méthode de Sugiyama, Tawaga et Tada [28] comme l'exécution successive de trois grosses étapes. Pour chacune de celles-ci, nous allons brièvement expliquer une piste de solution permettant d'obtenir, à la fin les coordonnées des différents noeuds optimisées pour un affichage le plus lisible possible. Les trois étapes sont les suivantes :

- Découpe du graphe en niveaux
- Arrangement des noeuds dans un même niveau
- Assignation des coordonnées.

5.4.1 Découpe du graphe en niveaux

Vu le caractère hiérarchique du graphe, il est évident qu'une des trois étapes serait la découpe en niveau du graphe. Pour ce faire, nous devons assigner à chaque noeud du graphe un rang. Les noeuds ayant obtenus le même rang formeront un niveau du graphe orienté à afficher. Le calcul du rang peut s'effectuer grâce à une des deux méthodes citée dans la section 5.3.5.

5.4.2 Arrangement des noeuds dans un même niveau

La première étape nous fournit pour chaque niveau un ensemble de noeuds. Ici, notre but va être d'ordonner cet ensemble. En effet comme on peut le voir sur la figure 5.4, un bon ordonnancement permet une meilleure lisibilité du graphe.

Ordonnancement lisible

Ordonnancement moins lisible

FIG. 5.4 – *Ordonnancement des noeuds dans un niveau*

Pour ce faire,[24] propose une méthode se basant sur l'affirmation suivante : L'ordre des noeuds dans tous les niveaux situées entre la racine et le niveau courant est déjà fixé. L'opération restant à accomplir sera d'ordonner le niveau courant avant de passer aux suivants. L'ordonnancement de celui-ci va être effectué grâce à un tri des noeuds. On va assigner à chaque noeud un poids. Celui-ci étant calculé grâce une fonction heuristique. L'idée de base étant qu'un noeud de niveau inférieur doit être quelque part au milieu des noeuds lui servants de père. De cette constatation, son poids va être fonction de la position de ses pères dans le niveaux supérieurs et de leur cardinalité. Le rang final du noeud dans le niveau sera proportionnel au score de la fonction.

Il est à noter que cette méthode ne tient pas compte d'un graphe représenté avec un minimun de croisements à l'affichage. Si on veut minimiser ceux-ci, il faut connaître une information complémentaire qui est le nombre de croisements vu l'ordonnancement du niveau. En permuttant certains noeuds, cet indice pourrait diminuer. Pour que le calcul soit pertinent, il faut effectuer la même opération pour les niveaux supérieurs et inférieurs jusqu'à ce que le nombre de croisements globaux soit minimun.

5.4.3 Assignation des coordonnées

Maintenant que les différents niveaux sont ordonnées, il ne reste plus qu'à assigner à chaque noeud du graphe leurs coordonnées d'affichage. La composante en "Y" est fixée par le numéro du niveau dans lequel se trouve répertorié le noeud considéré. Pratiquement, une distance égale est utilisée entre chaque palier du graphe. Par contre le calcul de l'abscisse reste à déterminer.

Entre les noeuds de chaque niveau, il ne faut pas oublier de respecter une distance minimale. Pour respecter cette condition et, en plus, assigner les abscisses des différents noeuds, [24] propose également une méthode d'assignation : la méthode du pendule.

Méthode du pendule

Dans cette méthode, nous utiliserons la méthaphore du pendule. C'est-à-dire que nous allons désigner un noeud comme étant le point d'ancrage d'un pendule. Ses fils seront les boules du système et les arcs reliants les occupants des différents niveaux seront des cordes. Le comportement naturel d'un tel système se termine par un état ou toutes les boules sont immobiles et leur placement est centré par rapport au point d'encrage. Cette propriété va être exploitée dans cette méthode. Pour chaque boule, on va lui assigner un poids. Ce poids est fonction de sa distance par rapport à la position du père. Si la boule se trouve à droite, alors le poids sera négatif. Sinon il sera positif. Le système sera en équilibre lorsque la somme des poids sera nulle. Pour arriver à cet équilibre, l'abscisse des noeuds inférieurs va être modifiée. On peut en voir une illustration à la figure 5.5. Un déplacement d'un ensemble de noeuds peut provoquer la migration de noeuds placés plus à droite dans le niveau. En effet il ne faut pas oublier qu'il existe une distance minimale entre chaque noeud. Si ce cas se présente, alors les noeuds pères devront aussi bouger. La phase d'assignation s'arrêtera lorsque toute la structure sera en équilibre.

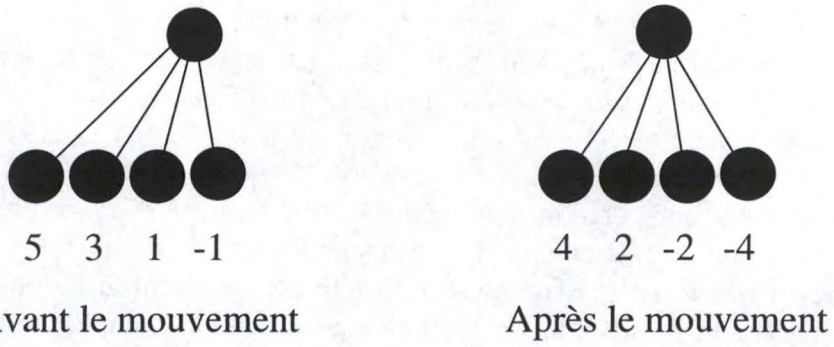

FIG. 5.5 – *Méthode du pendule*

5.5 Conclusion

Dans ce chapitre, nous avons passé en revue les plus grandes catégories de dessin de graphe. En analysant la situation, nous avons trouvé certaines pistes intéressantes pour la visualisation du graphe temporel de notre application.

Toutefois, une sémantique particulière est associée à ce graphe. De ce fait certaines contraintes de visalisation on été imposées comme par exemple :

- Les arcs représentants les différents médias seront placés horizontalement à l'écran.

- La longueur des arcs à l'écran sera proportionnelle à la durée des médias associés à ceux-ci..

Pour afficher un tel graphe, un seul moyen est possible: il faut dupliquer un noeud si plusieurs arcs sortants partent de celui-ci. Généralement les algorithmes associés à ces méthodes ne traitent pas ce cas de figure. De plus, dans la plupart des cas, un placement lisible des noeuds ne se soucie pas de la longueur des arcs à respecter.

Nous pouvons donc en conclure que nous ne pouvons pas appliquer ces méthodes telles quelles. Pour ce faire, nous devons les améliorées et les rendre aptes aux spécifications énoncées plus haut. Nous avons donc choisi de présenter dans le chapitre suivant une méthode satisfaisant aux critères imposés par l'affichage du scénario désiré.

Malgré le choix effectué, nous pourrons quand même faire appel à une des parties des autres méthodes citées auparavant comme par exemple le tri des noeuds dans un niveau, la méthode du pendule ou bien une autre référencée dans [7].

Chapitre 6

Visualiation du graphe temporel dans la vue Time Line

6.1 Introduction

Dans ce chapitre, nous allons découvrir la solution élaborée par nos soins. Elle s'éloigne des courants classiques évoqués dans la section 5.3 à la page 58. Cette divergence se justifie à cause de la sémantique spéciale associée au graphe temporel de MADEUS. Les besoins propres de la présentation du scénario imposent cette sémantique particulière. Nous allons donc énoncer ici les plus pertinents :

- La longueur des médias et des délais à l'écran doit être proportionnelle au poids des arcs du graphe temporel. En effet le poids de arcs représente la durée de présentation de ceux-ci.
- Les médias et les délais doivent être affichés horizontalement.
- Pour palier à la condition précédente, tous les noeuds du graphe comportant plus d'un arc entrants ou bien sortants doivent être dupliqués à l'affichage. Le nombre de copies va dépendre du placement spatial des objets ayant une extrémité commune avec ce noeud. Ce nombre varira entre d'une part le nombre mininum d'arcs entrants et sortants et d'autre part la somme de ces deux données.
- Il ne peut pas exister de croisement entre les medias (ou bien les délais) à l'affichage. Le respect de la seconde propriété implique le respect de celle-ci, si les différents éléments ne se chevauchent pas.
- Les instants de début et de fin relatifs par rapport au commencement de la présentation commun à plusieurs objets doivent être placés sur une même verticale à l'écran. Ceci permet de représenter un placement sur une ligne du temps cohérent. Cette propriété résulte de la combinaison des propriétés un et deux.

Cette liste de critères établie, nous allons dans la suite présenter les méthodes retenues pour dans un premier temps afficher le document et dans un second

améliorer la présentation de l'affichage suivant deux optiques. La première permettra de tracer un graphe lisible sur un espace d'affichage minimun. La seconde fournira un moyen de minimiser les croisements entre la représnetation des médias et celle des contraintes (un exemple de croisement est présenté à la figure 6.3 à la page 74). Nous terminerons par tester si la combinaison des deux méthodes est possible afin de donner une représentation optimale aux utilisateurs de MADEUS. Pour réaliser ceci, nous commencerons tout d'abord par récolter les données indispensables au tracé d'un graphe temporel respectant les crières énoncés plus haut.

6.2 Données utiles

Nous allons voir dans cette section quelles informations il est pertinent d'avoir dans la vue Time Line pour pouvoir visualiser l'information temporelle dont nous avons besoin. En pratique cela revient à déterminer ce que le document étendu de la vue doit absolument contenir. Nous allons dans la suite voir aussi où et comment récupérer ces informations.

6.2.1 Copie du document original

Pour constituer le squelette du document étendu, nous allons tout d'abord copier le document de référence. Cette copie, est une copie récursive, c'est-à-dire que le document va être copié ainsi que tous les documents compris dans celui-ci. A la fin de cette opération, les deux documents seront totalement distincts et autonomes. Maintenant, on va pouvoir ajouter les informations propres à la vue.

6.2.2 Ajout de l'information temporelle

Nous allons récupérer certaines informations temporelles. Pour chaque arc pertinent du graphe apppartenant au document (c'est-à- dire les arcs représentants les médias et ceux contenant les delais qui vont être affichés), nous allons leur ajouter un champ exprimant sa longueur à l'affichage. Pour connaître cette information, il suffit de récuperer la valeur préférée du temps de présentation de l'arc. En multipliant cette valeur par un facteur d'échelle, nous avons donc calculer la longueur de l'objet à l'écran. Pour avoir un affichage correct, il reste à déterminer où placer les différents objets sur la surface d'affichage.

6.2.3 Récupération des coordonnées d'affichage

Nous allons dans cette partie voir comment à partir du graphe temporel il est possible de récupérer des informations spaciales pour l'affichage des médias et des arcs. Pour ce faire, nous avons en premier lieu choisi de visiter le graphe grâce à

un parcours en profondeur d'abord. Cette technique nous permet de passer par tous les arcs du graphe sans en oublier un seul. Pendant le parcours, plusieurs chemins vont être empruntés. A chaque chemin passant par un arc non encore visité, va être assigné un numéro de ligne. L'ordonnée des différents objets à afficher sur l'écran sera proportionnelle à ce numéro. De ce fait, nous pouvons assurer que deux elements ne se chevaucheront pas sur une même ligne. Nous venons de trouver la composante des ordonnées pour chaque élément. La détermination de l'autre composante est effectuée au fur et à mesure de l'avancement dans le parcours. La valeur de présentation préférée d'un document étant aussi proportionnelle à la longueur préférée d'un arc, il suffit pour connaître le positionement d'un arc sur l'axe des abscisses de connaître son instant de début par rapport au noeud de départ. Chaque arc a comme instant de début, la valeur de l'instant de fin de l'arc le précédant dans le parcours. Pour connaître cette composante, il suffit donc chaque fois que l'on avance d'un arc dans le parcours, de spécifier l'instant de début de l'arc comme étant l'instant de début de son précedent augmenté de la durée préférée de celui-ci. Evidemment, nous devons aussi multiplier les données récoltées par le même facteur d'échelle que celui employé pour calculer la longueur à l'écran des rectangles représentants les objets. Certaints éléments de la présentation sont de type composite. Il ne faut pas oublier d'effectuer les mêmes opérations pour les graphes représentants leurs contenus. Cette opération effectuée une fois pour toute, on pourra par la suite les ouvrir et les fermer sans se soucier à nouveau du probleme de l'affectation des coordonnées de leur placement. Il faut encore réfléchir à une petite chose : les coordonnées sont exprimées pour un document principal. C'est-à- dire que le premier arc sera dessiné à l'emplacement (1,1). C'est pourquoi il ne faut pas oublier d'ajouter une information supplémentaire à tous les objets : le placement du père. De ce fait, on n'aura pas de chevauchement si plusieurs composites sont ouverts en même temps.

6.3 Simple Affichage du document

Une fois les différentes informations récoltées, le travail restant à effectuer pour l'affichage est très simple. Il suffit d'afficher à l'écran les différents documents à présenter. Pour ce faire, nous allons parcourir un à un les documents du composite principal, identifier leur type et afficher le graphique correspondant. Rappelons que sa taille est déjà connue auparavant. Ses coordonnées d'affichage sont données par l'addition de son placement relatif et des coordonnées du composite l'englobant.

On remarque qu'avec la technique de parcours du graphe et d'assignation de l'ordonnée des différents éléments, il peut se poser le problème suivant. Le document s'étend sur beaucoup plus de ligne que ce qui est voulu. Cela donne une vue d'ensemble moins bonne de la présentation comme on le voit bien sur la figure 6.1. Si le document est composé de plusieurs scènes s'enchaînant les unes

après les autres, l'affichage du scénario s'étendrait sur une hauteur inconsidérable. Avec un "compactage" de celui-ci, on pourrait réduire drastiquement le nombre de lignes à présenter dans la vue. Dans la section suivante, nous allons voir comment remédier à ce petit inconvéniant.

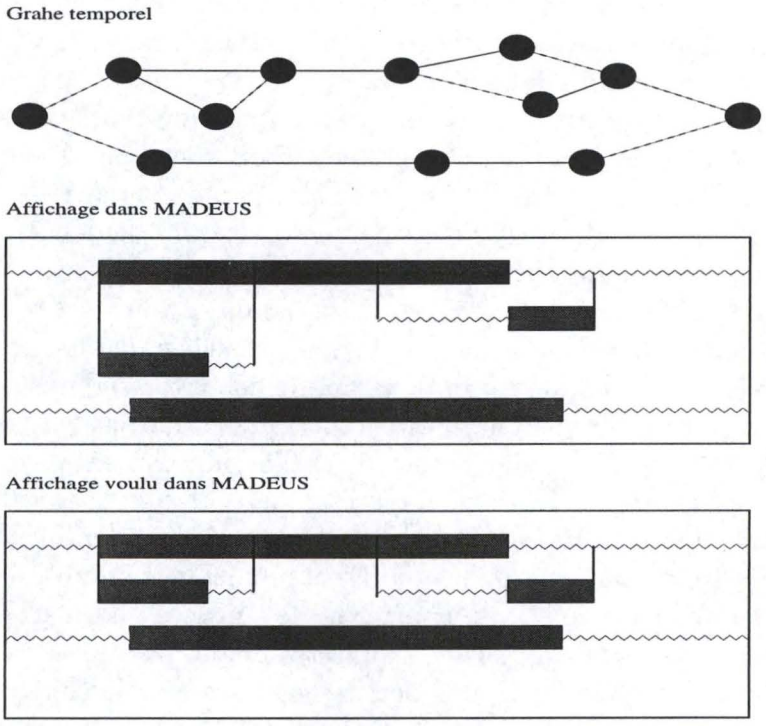

FIG. 6.1 – *affichage temporel dans la vue Time Line*

6.4 Affichage plus compact du document

Pour effectuer cette opération nous allons nous baser sur les propriétés du parcours d'arbre utilisé. On peut remarquer que chaque ligne définie lors de la récuperation des informations peuvt être représentée par un seul intervalle continu. Sa borne inférieure étant le moment de début du média ayant l'abscisse la plus faible, quant-à sa borne supérieure elle correspond à la fin du média ayant l'abscisse la plus élevée augmentée de sa taille. On peut définir pareil intervalle pour chaque ligne. Ces lignes vont avoir les propriétés spatiales suivantes.

- La première ligne sera une ligne couvrant l'entièreté de la présentation
- La borne supérieure d'une ligne inférieure (si les lignes ont une numérotation croissante) n'est pas plus petite que la borne inférieure d'une ligne supérieure.
- Il est possible que la borne supérieure d'une ligne supérieure soit plus grande que la borne supérieure d'une ligne inférieure.

Le but de cette section est de voir quelle méthode peut être appliquée pour "compacter" l'affichage. Nous allons procéder en quatre étapes.

6.4.1 Détermination du degré maximun de liberté par rapport aux lignes supérieures

Pour commencer, nous allons tout d'abord définir le concept de degré de liberté pour une ligne. Dans ce contexte, le degré de liberté d'une ligne est le nombre maximun de lignes séparant sa position actuelle et la position la plus haute que cette ligne peut atteindre. On va fixer le degré de liberté de la première ligne à 0. Pour les autres, nous devons comparer le positionnement d'une ligne par rapport à celles placées au-dessus d'elle. Si l'intervalle représentant la ligne peut s'insérer dans la ligne précédente sans chavaucher le sien, alors on peut augmenter le degré de liberté d'une unité. On répète cette opération jusqu'a ce que l'on ne puisse empêcher le chevauchement. L'autre possibilité est d'avoir la borne supérieure de l'intervalle de la ligne courante plus grande que la borne supérieure d'une ligne inférieure. Tant que cette configuration est réalisable, on augmente le degré de liberté de la ligne.

Cette étape permet de donner un degré de liberté relatif par rapport à la ligne la plus basse limitant le déplacement vertical. Or cette ligne peut bouger aussi. Dans ce cas, le degré de liberté sera la somme de ces deux parties. Une illustration du déroulement de cette première étape est proposée sur figure 6.2.

6.4.2 Ajustement du degré de liberté par rapport aux lignes supérieures

Lors de la première étape, il se peut que le degré de liberté donné soit plus grand que dans la réalité. Ceci se produit dans le cas où une ligne supérieure à la ligne considérée possède un degré de déplacement plus petit. On peut voir un exemple à la figure 6.2 avec les lignes 6 et 7. Nous ne pouvions pas détecter ce genre de limitation dans la première étape, car nous faisions une analyse des lignes dans un ordre croissant. Pour remédier à cet inconvénient, nous allons procéder dans le sens inverse. Si une ligne a un plus petit degré de liberté qu'une ligne inférieure, que sa borne inférieure est supérieure ou égale la borne inférieure de la ligne considérée et que son degré de liberté est non nul, nous allons diminuer le degré de la ligne considérée de la valeur du degré de liberté de la ligne bloquante. On peut aussi suivre la totalité de cette étape à la figure 6.2 étape 2.

6.4.3 Détermination du degré de liberté par rapport à la première ligne

Il reste encore à traiter le cas des degrés de liberté nulle et à mettre aussi à jour l'information des lignes bloquantes trouvées dans la seconde étape. Pour ce faire, dans cette étape, nous allons repartir de la deuxième ligne et descendre pour mettre à jour toutes les lignes. On va donner à une ligne le degré de liberté de la ligne qui la précède dans le cas ou celle ci a sa borne inférieure supérieure ou égale à la borne inférieure de la ligne précédente. Si ce n'est pas le cas, c'est que nous nous trouvons sur un autre palier. On peut voir ceci sur la figure 6.2 entre les lignes 3 et 4 ou bien entre les lignes 5 et 6.

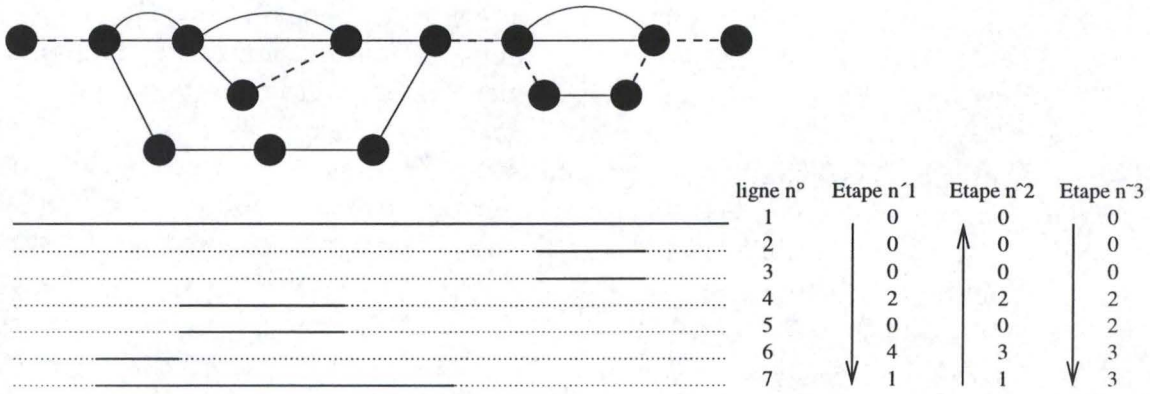

FIG. 6.2 – *différentes étapes du "compactage" de l'affichage*

6.4.4 Modificataion des données dans le document

Dans les étapes précédentes, nous avons seulement déterminé le degré de liberté de chaque ligne. Il faut encore répercuter les informations aux differents composants des lignes qui doivent bouger. Pour ce faire, tous les éléments d'une ligne vont avoir leur ordonnée diminuée du degré de liberté correspondant à la ligne les hébergeants.

6.4.5 Modification de l'affichage

Pour compléter l'opération, il suffit de rafraîchir l'affichage pour tenir compte des modifications apportées aux coordonnées.

Remarque : il est évident que nous devons effectuer la même opération pour tous les éléments contenus dans les autres composites de la présentation afin de garder une certaine cohérence dans la méthode d'affichage.

6.5 Affichage du document minimisant les croisements

La visualisation du scénario temporel dans MADEUS implique un double affichage. Il faut dans un premier temps faire apparaître à l'écran le dessin du graphe temporel remanié et dans un second la représentation graphique des differentes contraintes. Le parcours d'arbre utilisé pour décider du placement des objets ne tient pas compte de cette seconde composante. Il se peut dès lors que la superposition des lignes verticales représentant les contraintes coupent un objet affiché précédemment. Le but de cette section est d'essayer de minimiser les coupures à l'écran pour donner à l'auteur une meilleure perception du scenario en lui proposant un affichage plus "lisible". Pour ce faire, nous allons présenter une méthode en plusieurs étapes qui pourrait être appliquée.

6.5.1 Découpe en niveau du graphe

La première étape consiste à créer en regard du graphe temporel une structure hiérarchique de celui-ci. C'est-à-dire une structure reprenant la découpe en niveau de ce graphe. Les différents niveaux seront détermines par l'instant de début associé à un arc (valeur préférée de son temps de présentation). Autrement dit, les niveaux de hiérarchie seront déterminés par le positionnement temporel des differents noeuds du graphe. Un niveau pouvant inclure plusieurs noeuds. Pour chaque niveau défini, la structure comprendra un pointeur vers les différents documents associés aux médias ou bien aux délais affichables. En passant par cette structure, il sera donc aisé de modifier les coordonées d'affichage des arcs du graphe temporel.

6.5.2 Recherche des croisements

Une fois la structure hiérarchique construite, nous pouvons commencer la recherche des croisements. Il y aura un croisement lorsque pour au moins deux arcs ayant un noeud en commun, leur ordonnée ne se trouve pas sur des lignes adjascentes et entre ces lignes respectives est venu s'intercaler un autre arc dont les noeuds matérialisant ses extrémités n'appartiennent pas au même niveau de hiérarchie que le noeud cité ci-dessus. Un exemple est présenté à la figure 6.3 Concrètement pour retrouver un croisement, il suffit de parcourir la structure hiérarchique niveau après niveau. Pour chacun d'eux, il faut vérifier l'absence de croisements avant de passer au suivant. Si ce n'est pas le cas, alors nous devons l'éliminer.

Dans la pratique, nous allons construire une matrice avec un nombre de lignes égales aux nombres de lignes sur lesquelles s'étend l'affichage et un nombre de colones égales aux nombres de niveaux de la structure hiérarchique. Les différents

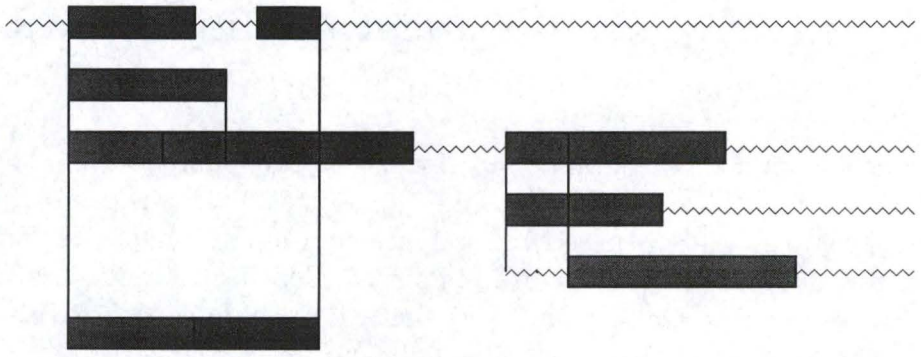

FIG. 6.3 – *Croisements entre médias et relations dans la vue Time Line*

éléments de la matrice seront des noeuds dont la répartition pour la colone (c) sera la suivante : si il existe pour une ligne d'affichage donnée (l) un noeud appartenant à la structure hiérarchique au niveau (c), alors l'élément (l,c) de la matrice sera ce noeud, sinon on prendra l'élément (l,c-1). Ceci est evidemment possible si la première ligne de la matrice est remplie complètement. Cela implique la possibilité de présence d'un valeur UNDEF. Dans le repérage, la valeur UNDEF jouera le rôle d'un élément neutre.

6.5.3 Détermination de la racine

Le croisement peut être matérialisé comme une mauvaise assignation des coordonnées spatiales des différents documents à affiché. On peut aussi le voir comme un mauvais choix effectué lors du parcours initial du graphe temporel, celui-ci se pursuivant par une branche non voulue. En obligeant l'ordre du parcours, nous pourrons modifier les positions spatiales.

Pratiquement ceci n'est possible que si on trouve la racine commune aux chemins joignants les différents noeuds impliqués dans le croisement à un point de rencontre commun si on se déplace dans le sens inverse de l'orientation du graphe. Ce point sera le noeud a partir duquel le choix de l'arc suivant dans le parcours a été erronné.

Il faut retrouver la racine commune aux éléments en croisement pour savoir à partir d'où on pourrait modifier le parcours primaire de l'arbre et de ce fait avoir le résultat voulu à l'écran. On va donc travailler sur les deux noeuds en croisement. En suivant l'ordre inverse de la hierarchie, nous allons remonter chaque arc entrant à ces noeuds pour établir le/les noeuds qui les précède(nt). Si tous les chemins empruntés arrivent au même noeud, alors on a trouvé la racine, sinon il faut continuer la progression dans le sens inverse de l'orientation du graphe.

A la figure 6.4, nous montrons en pratique la détermination de la racine. Nous utilisons la matrice créée à la seconde étape. Sur le dessin, les U composants la matrice sont simplement l'abréviation de la valeur UNDEF.

6.5.4 Détermination de la partie à bouger

Une fois la racine trouvée, nous avons le point de départ duquel on devra effectuer des changements de coordonnées. Il reste encore à déterminer quels objets vont être déplacés. Ces objets seront compris dans un sous graphe partant d'un ou plusieurs arcs sortant de la racine et se prolongeant peut-être au dela des noeuds en conflits. Par construction de la structure d'affichage, tous les objets à bouger se situerons dans un intervalle formé par les deux objets accrochés aux noeuds subissant le croisement. L'objet possédant l'ordonnée la plus petite détermine sa borne inférieure. Tandis que l'objet ayant quant-à lui l'ordonnée la plus grande donne la borne supérieure. On remarque que lors de la phase de recherche de la racine, les premiers mouvements étaient tous effectués à l'interieur de cet intervalle. En repartant des noeuds identifiés à l'étape précédente, nous allons nous rediriger vers la racine. A ce moment, tout élément appartenant à un chemin passant par l'arc initiateur du croisement est compris dans le sous-graphe contenant les objets à deplacer.

En pratique, nous nous déplacerons dans la matrice pour retrouver la structure mobile. Pour chaque ligne, nous allons calculer un intervalle de présence. Cet intervalle va donner la largeur exprimée en ligne de la structure solidaire à l'objet de départ se trouvant sur une ligne d'affichage donnée(ligne A). Si le noeud de départ de cet arc possède par exemple un autre arc sortant sur la ligne B, alors l'intervalle de la ligne A va être fusionné à celui de la ligne B. Le nouvel intervalle va être assigné aux deux lignes. Si une ligne matérialise un noeud répertorié UNDEF dans la matrice, alors l'intervalle assigné à cette ligne sera la copie conforme de la ligne inférieure. A la fin, toutes les lignes ayant le même intervalle que celle contenant l'objet provoquant le croisement devront être déplacées dans l'affichage. On peut en voir l'application à la figure 6.4

6.5.5 Mise à jour des coordonnées

Il reste encore à assigner à tous les arcs du sous-graphe se déplaçant à l'écran leur nouvelles coordonnées de positionnement spatiales dans la vue. Il ne faut pas non plus oublier de changer celles de tous les éléments partant de la racine commune. Comme de même, il faudra mettre à jour les informations concernant l'affichage des relations à l'écran.

En pratique, nous allons tout d'abord décider du nouveau placement du sous graphe. Par convention si l'intervalle contenant la structure à déplacer contient la première ligne de l'intervalle délimitant les calculs (appelons-le LIM), alors nous placerons ces objets en dessous des autres. Sinon, nous les placerons au-dessus de ceux-ci. Comme représenté à la figure 6.5

Pour mettre à jour toutes les coordonnées, nous allons nous servir de la structure hiérarchique. Il suffit de commencer au niveau contenant la racine. Pour chaque objet étant contenu dans l'intervalle LIM, de modifier ses informations

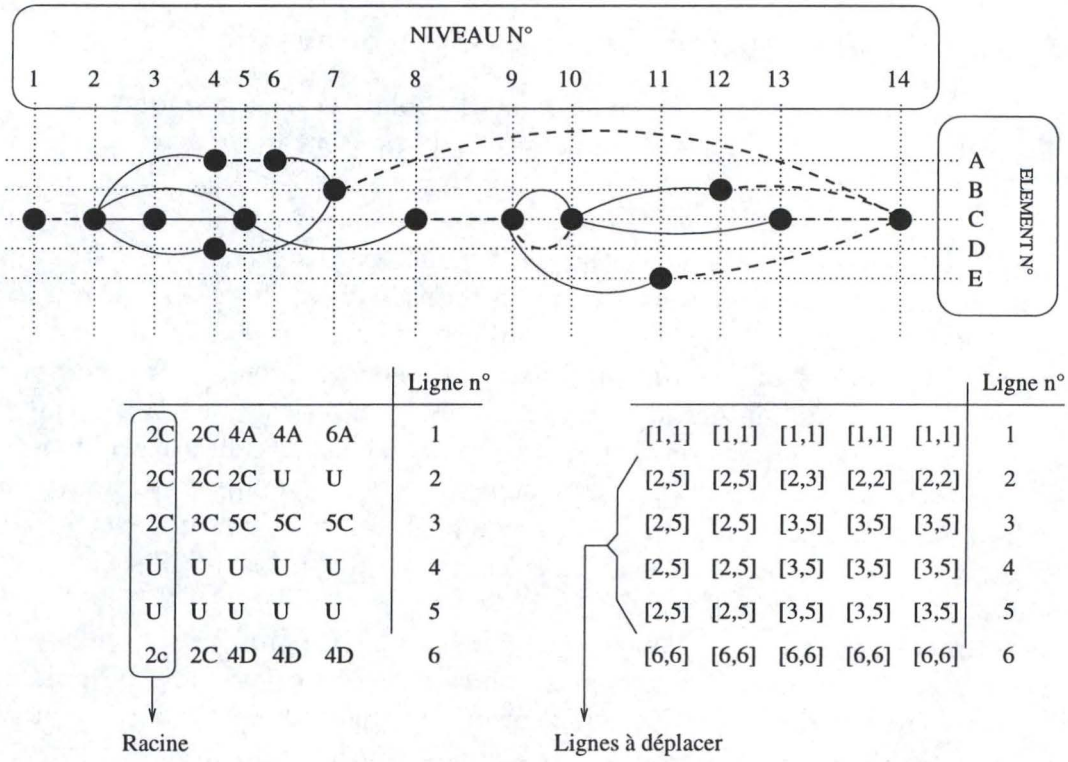

FIG. 6.4 – *Etapes trois et quatre de l'élimination des croisements*

concernant son ordonnée à l'écran. Cette opération sera répétée pour chaque niveau supérieur à la racine.

6.6 Affichage compact minimisant les croisements

On remarque bien dans la section précédente que la minimisation des croisements prend une place assez importante à l'affichage. Nous allons donc voir si on pourrait utiliser la méthode de "compactage de l'affichage vue à la section 6.4. Puisque dans la minimisation on ne fait que remonter des lignes, il faut vérifier si les propriétés des lignes définies antérieurement sont elles aussi vérifiées. Dans ce cas, l'application telle quelle de cette méthode sera possible.

- La première n'est peut être pas toujours vérifiée. Mais le principe de base reste applicable quand même, car on fixe le degré de liberté de cette ligne à 0.
- Les différentes modifications de place sont effectuées à hauteur d'une racine commune. Ceci empêche donc une violation de la seconde propriété.
- La troisime propriété reste d'application.

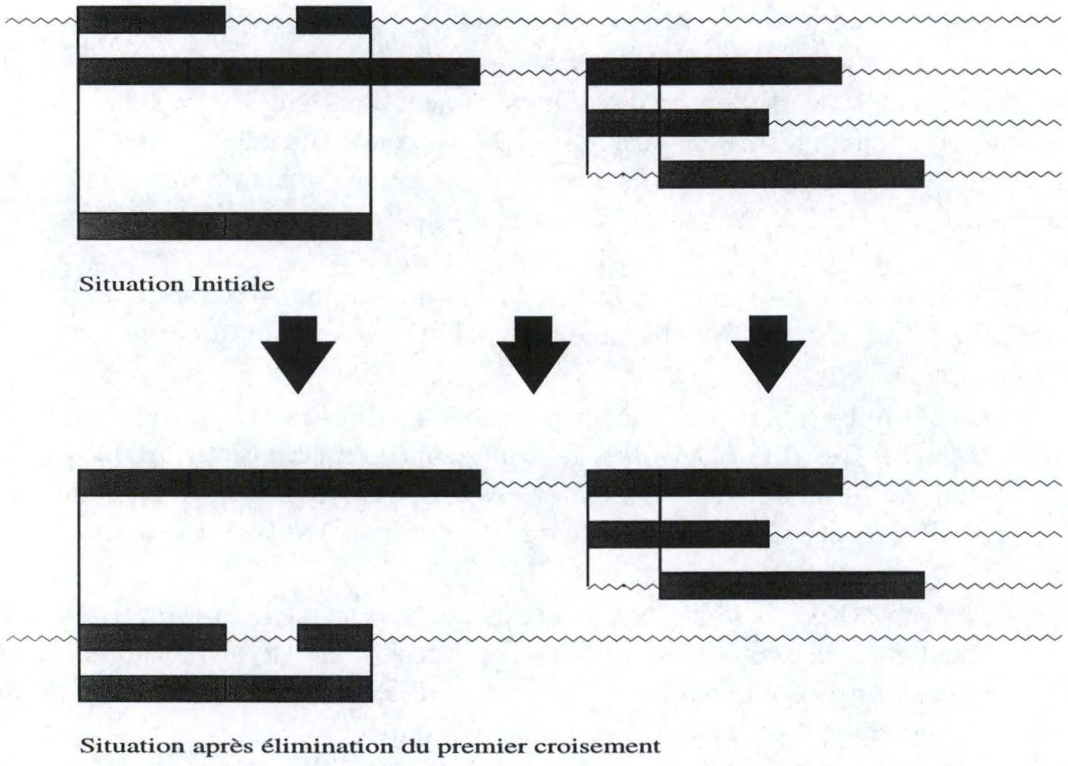

FIG. 6.5 – *Modificaton des coordonnées des divers éléments*

Les propriétés de départ permettent d'appliquer la méthode vue précedemment. Les conditions de rehaussement restant les mêmes, l'opération globale sera seulement une application séquentielle des principes exposés dans la section 6.5 (page 73) enchaînés à ceux expliqués dans la section 6.4 (page 70)

6.7 Conclusion

Dans ce chapitre, nous avons présenté une méthode d'affichage du graphe temporel suivi de trois fonctions d'améliorations de celle-ci. Dans la pratique, ces différentes fonctions devraient être applicables independamment l'une de l'autre selon la décision de l'auteur. Son choix peut être motivé par les critères suivants :

- Si le document est divisé logiquement et structurellement en plusieurs scènes se suivant l'une après l'autre, le compactage de l'affichage nous paraît une méthode assez intéressante.
- Si un composite du document comporte beaucoup d'objets. Que ceux-ci sont reliés par une multitude de contraintes, alors dans ce cas il est intéressant de minimiser les croisements afin d'améliorer la lisibilité du scénario.
- Dans un cas reprennant toutes les caractéristiques citées ci-dessus, nous

opterons pour un affichage compact minimisant les croisements.

En examinant cette petite analyse, nous pouvons dériver le comportement à proposer par défaut lors de l'ouverture d'un document dans MADEUS. Généralement, les compositions multmédias sont divisées en scènes (un outil comme DIRECTOR vu à la page 21 dans la sous-section 2.4.2 intègre même une vue spéciale permettant la gestion des scènes d'un scénario). De ce fait, un compactage de la représentation du document semble un bon compromis.

Il nous reste encore dans cette conclusion à examiner l'apport potentiel des améliorations apportées à l'affichage pour l'édition de la composante temporelle du document.

- La minimisation de croisement permet d'améliorer la visualisation des contraintes. De ce fait, le repérage d'une contrainte à l'écran est facilité. En effet, si les lignes vertitales matérialisants une partie d'une contrainte sont rectilignes et non croisées, la charge cognitive de l'auteur est diminuée pour cette opération.
- Le compactage de l'affichage assure à l'auteur une meilleure facilité d'édition. En effet, sans ce mécanisme, l'auteur pourrait mal dicerné pourquoi certains délais sont pris en compte dans le calcul de l'intervalle de retaillage vu leur "éloignement" du média en cours de traitement.
  De plus un affichage non compact implique souvent une utilisation des ascenceurs associés à la vue. Cette opération serait soit superflue, soit diminuée dans la version compacte. Un petit détail comme celui-ci peut inciter l'auteur à utiliser notre outil ou bien à le délaisser si il ne le trouve pas adapté à ses besoins.
- Le compactage permet dans une mesure minime de traiter le problème de la mise à l'echelle, mais il n'est bien entendu pas sufisant pour l'éliminer.

Chapitre 7

Perspectives

7.1 Introduction

Dans ce chapitre, nous allons présenter des améliorations possible de l'outil d'édition MADEUS. Certaines paraissent pourtant triviales. Si nous les avons répertoriées dans ce chapitre, c'est à cause du statut de notre logiciel. En effet, MADEUS est toujours en phase de développement. Le stage se terminant en janvier, il se pourrait aussi que des propositions décrites ci-dessus soient déjà intégrées dans l'outil.

Ces perspectives concernent la visualisation et l'édition de la partie temporelle du document. Pour commencer, nous allons proposer un placement personnalisé des objets dans la vue Time Line et une représentation particulière des hyperliens. Pour terminer ce tour d'horizon, nous poserons le délicat problème de l'édition de contraintes.

7.2 Placement personnalisé des objets dans la vue Time Line

Comme nous l'avons déjà vu au chapitre 6, le placement des différents objets dans la vue Time Line est "imposé" lors du parcours du graphe temporel. On retrouve le même phénomène dans le cas d'une élimination des croisements. Dans cette section, nous voudrions présenter une alternative à ce placement automatique. Une fois le document affiché dans la vue, nous allons donner la possibilité à l'auteur de décider lui-même de la position des différents objets. Pour ce faire, nous allons procéder en plusieurs étapes.

7.2.1 Sélection du média à bouger

La première chose est de voir si un média peut être bougé ou non. Un média peut changer de place à la condition suivante. Dans le graphe temporel, le noeud de début auquel est connecté l'arc réprésentant le média doit avoir une liste de noeuds sortants dont la cardinalité est supérieure à un. C'est le seul cas envisageable ici. Ceci nous impose une petite limitation. Citons par exemple le cas d'une contrainte A FINISHES B, dans ce cas si on veut bouger A, on devra sélectionner le délai encapsulé entre les deux médias.

Avec cette convention, le problème se ramène plus ou moins à la même chose que l'élimination des croisements. Dans notre cas ici, l'élément sélectionné de par sa position détermine directement la racine. Les lignes que nous devons bouger sont toutes celles comprises entre la ligne contenant l'objet à bouger et la ligne matérialisant un autre arc sortant de la racine placée plus bas sur l'affichage que la premiere citée. Cette dernière ne faisant pas partie du lot à bouger. Cette affirmation est verifiée si bien entendu la ligne à bouger n'est pas la dernière. Dans le cas contaire, nous nous référerons à l'étape 4 présentée à la sous-section 6.5.4 du chapitre 6.

7.2.2 Limitation du déplacement

Les bornes possibles du déplacement sont très faciles à calculer. Il suffit de connaître pour tous les arcs partants directement de la racine leur ordonnée à l'ecran. La borne inférieure sera l'ordonnée la plus petite de ces rectangles diminuée de la moitié de la distance entre deux lignes. La borne supérieure sera quant-à elle l'ordonnée la plus grande des rectangles partants de la racine, mais cette fois-ci augmentée de la valeur citée précédemment. Cette augmentation de taille de l'intervalle permet de visualiser à l'écran un placement au-dessus ou en-dessous de toute la structure définie. Comme cela, l'auteur à bien l'impression de mettre les objets au dessus/ en dessous des deux lignes limitrophes. Si l'objet à deplacer est contenu sur une de ces deux lignes, l'intervalle sera bien entendu réévalué pour exclure les différentes lignes à bouger.

7.2.3 Modification de son ordonnée

Dans cette étape, nous allons bouger la partie voulue. Pour ce faire, une bonne solution est de surimprimer à l'écran une structure en fil de fer de la partie déplacée. L'intervalle défini à l'étape précédente permet alors de limiter en largeur le déplacement. Une fois la nouvelle place choisie, il ne reste plus qu'a mettre à jour les coordonnées d'affichage.

7.2.4 Mise à jour des coordonnées d'affichage

Cette étape se déroule exactement de la même façon que l'étape 5 présentée à la sous-section 6.5.5. Il suffit alors de réafficher le doucment. Cette opération est décrite dans la section 6.3 du même chapitre.

7.2.5 Remarque

Le fonctionnement de MADEUS implique que la détermination des coordonnées spatiales s'effectue lors de l'ouverture de la vue Time Line. Il en va de même à chaque fois que l'on change de document. Ceci implique qu'à chaque fermeture du document, les modifications apportées par l'utilsateur sont perdues. Pour remédier à ce problème, il faut simplement modifier la DTD MADEUS. En ajoutant un attribut optionnel pour chaque média qui donnerait pour la vue les coordonnées à utiliser. On pourrait penser à un tag du genre TLX="100" TLY="25" ou TLX désignerait la position relative de l'abscisse du média à l'ecran et TLY l'ordonnée de celui-ci. Les positions relatives s'expliquent par la possibilité d'ouvrir un composite. Les composites secondaires, peuvent aussi avoir subit des modifications. Dans ce cas, avec des coordonnées relatives, nous n'avons plus de problème les concernant. Ce mécanisme ne résoud pas tous les problèmes. Il reste encore à donner aux délais une localisation à l'écran. Pour ce faire, on pourrait imaginer que dans la déclaration d'une contrainte temporelle, l'éditeur ajoute lui-même pour chaque délai impliqué son identifiant propre qui sera utilisé à chaque fois et ses coordonnées spatiales. Ces attributs étant aussi optionnels. Voici un exemple de cette nouvelle syntaxe possible :

```
<Relation>
<Temporal>
<DURING intervalles1 = "toto" intervalles2="titi"
        TLDELAY="1220 100 50" TLDELAY="1221 220 50">
</Temporal>
</Relation>
```

La valeur de l'attribut TLDELAY comprend les trois informations importantes qui sont dans l'ordre l'identifiant du noeud, l'abscisse à l'écran et l'ordonnée à l'écran.

7.3 Visualisation des hyperliens

Dans cette section, nous allons nous concentrer un instant sur la visualisaton des hyperliens. Un hyperlien est un lien qui doit être activé par l'utilisateur. Il existe deux sortes d'hyperliens dans MADEUS. Tout d'abord nous allons nous concentrer sur les liens intra-documents.Dans la suite, nous parlerons de liens inter-documents.

7.3.1 Liens Intra-Documents

Description

Un lien intra-document est un lien créé entre deux objets du même document. Il est intéressant de pouvoir visualiser ce genre de liens dans une vue comme la Time Line. Généralement ces liens sont utilisés couramment à la fin suivante: L'hyperlien permet à l'utilisateur visionnant la présentation d'arrêter la scène en cours et d'arriver directement dans une autre partie du document dont l'instant de présentation peut se situer bien avant ou bien après le moment de l'invocation de l'activation du lien. On peut voir des applications de ce type sur les disques DVD. Au démarrage du disque, on présente un menu à l'utilisateur qui peut si il le désire voir le film de la première seconde jusqu'à la fin ou bien choisir les séquences préférées qu'il veut visualiser. La vue Time Line est bien adaptée pour représenter ce genre de choses. Elle contient tout le scénario temporel plus une ligne du temps. La combinaison de ces différentes informations permet à l'auteur d'avoir une bonne vue de l'interactivité proposée dans son document.

Représentation de l'hyperlien

Un hyperlien est matérialisé dans MADEUS par un média. C'est-à- dire que l'auteur peut potentiellement placer sur n'importe quel média un hyperlien. Donc pour differencier les objets ayant des potentialités de saut temporel, nous aimerions proposer dans la vue Time Line les conventions de représentation suivantes:

1. Les liens vont avoir une apparence visuelle propre.
2. Les médias visés par un lien vont également avoir une apparence visuelle propre.
3. On va essayer de montrer le plus possible la relation entre le lien et le média visé.

Pour ce faire, voici quelques propositions de représentations qui satisfont aux différentes conventions évoquées ci-dessus.

1. On pourrait superposer sur le rectangle représentant le média une double flèche. Cette représentation n'est pas mauvaise dans notre cas. Suivant l'orientation de la flèche, on peut distinguer facilement si le lien pointe vers une partie ultérieure ou bien une partie antérieure du document. Nous pensons qu'avoir une représentation différente pour les deux cas possibles de sauts temporels est une bonne chose. En effet, l'auteur peut d'un seul coup d'oeil vérifier le bon comportement d'un lien.
2. Pour différencier le lien du média à atteindre, nous avons pensé utiliser l'analogie pouvant être déduite entre d'une part le lien et l'objet visé et d'autre part entre une flèche et une cible. C'est pourquoi, pour représenter un objet référencé par un lien, on pourrait superposer sur le rectangle matérialisant celui-ci un dessin de cible. L'objet atteint, n'a besoin que

d'une seule convention de représentation comparée au lien. Cela n'apporte pas grand chose de connaître le sens d'approche du média.

3. La meilleure façon de mettre en évidence les liens serait de relier les objets en relation par une ligne continue traversant l'affichage. Cette ligne serait tracée dans une couleur non encore utilisée pour permettre une meilleure lisibilité du raccordement. Le problème de la lisibilité reste assez conséquent. On a déjà pu voir dans le chapitre 6 que l'on faisait tout ce qui etait possible pour minimimiser les croisements à l'affichage et de ce fait avoir un document agréable et lisible. Pour garder cette lisibilité, une bonne façon de faire serait d'afficher les connexions entre un lien et une cible seulement par une interaction de l'auteur. En effet, on pourrait ajouter au menu spécifique de la vue, une option permettant d'activer ou pas cette fonctionnalité. Dans le cas favorable à cette proposition, si l'auteur clique sur une partie du lien ou bien de la cible lorsque l'apparence du curseur sera modifiée comme le montre la figure 7.1, alors tant que le bouton de la souris n'est pas relaché, la connexion reste toujours visible. Pour éviter d'effectuer ces manipulations un grand nombre de fois, on peut aussi doter les paires de rectangles (lien/cible) d'un numéro unique par paire permettant d'identifier un lien par rapport aux autres comme montré à la figure 7.2. Ces différentes techniques permettent une bonne représentation des liens.

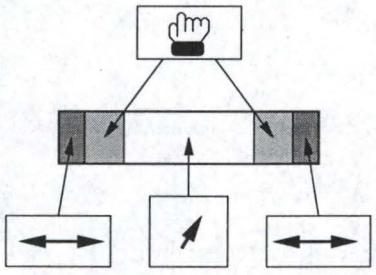

FIG. 7.1 – *Différentes formes du curseur*

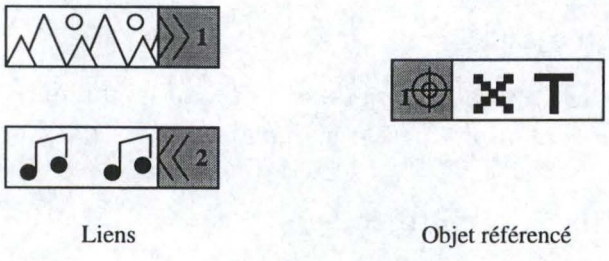

FIG. 7.2 – *Convention de représentation des liens intra-documents*

7.3.2 Liens inter-Documents

Description

Un lien inter-document est un lien créé entre deux objets de documents différents. Une application répandue et omniprésente dans la réalité est la navigation sur le word wilde web (WWW) à l'aide d'un browser. Dans MADEUS, grâce à cette sorte de lien, l'auteur peut créer un document stylisé avec une interface orientée web.

Représentation de l'hyperlien

Le développement exposé lors de la présentation des liens intra-document est valable aussi pour ce type de lien. Seul la représentation grahique changera. Sinon les différentes conventions de représentations et les propositions sont identiques. On peut voir un exemple de lien inter-document sur la figure 7.3.

FIG. 7.3 – *Convention de représentation des liens inter-documents*

7.4 Edition des contraintes temporelles dans la vue Time Line

Dans cette section, nous allons simplement poser les différents problèmes qui apparaissent lors de la réalisation de l'opération d'édition. Pour de plus amples informations, il est intéressant de parcourir [21].

7.4.1 Ajout d'une contrainte

Préparation de l'ajout

La sélection des différents objets se fait par la manière habituelle. Il suffit alors de choisir la contrainte à inserer dans la boîte à outil créée à cet effet.

Vérification de la faisabilité

Ces conventions de vérification vont être effectuées à un niveau très sommaire.

- Il ne doit y avoir que deux objets sélectionnés pour permettre l'ajout d'une contrainte temporelle..

- Les médias doivent être contenus dans le même composite.
- Il ne doit pas exister un chemin reliant les deux médias dans le graphe temporel associé au composite englobant.

Modification du graphe temporel

Suivant les sortes de contraintes à ajouter, cela va apporter certaines modifications dans le graphe associé au composite englobant

- Ajout potentiel d'un délai : relation before, after, starts, finishes
- Ajout de plusieurs délais : relation during, overlaps
- Fusion de noeud : relations equals, starts, finishes, meets.
- Suppressions de délais :
  Cette solution doit être envisagée si un média contenu dans le graphe n'est compris dans aucun jeu de contrainte. Il est simplement relié au noeud de début et au noeud de fin par des délais. Une fois le média impliqué dans une contrainte temporelle, ces délais sont susceptibles de disparaître. Les différents cas possibles sont assez complexes et variés pour voir ce qu'on doit ajouter et retirer. Par exemple, si on ajoute une relation START entre deux objets sans relation, il va y avoir suppression de deux délais et non de quatre.

Modifiation de la liste de contraintes

Pour que les opérations de visualisation et d'édition puissent se poursvuivre sans peine, nous devons mettre cette information à jour. Sinon, il serait potentiellement impossible de supprimer une contrainte ajoutée précédemment si celle-ci ne faisait pas partie de cette liste.

Modification de l'affichage

Comme le graphe temporel à été modifié, il faut vérifier à nouveau la cohérence temporelle. Si celle-ci n'est pas vérifiée, alors on doit soit prévoir un système d'UNDO pour revenir à la situation précédente, soit modifier les durées des objets automatiquement, mais bien sûr avec une confirmation demandée à l'utilisateur au préalable. Comme exemple, on peut prendre l'ajout d'une relation EQUALS entre des médias ayant des durées différentes dont les intervalles ne se chevauchent pas.

Une fois la cohérence temporelle assurée, nous devons réitérer les opérations énoncées dans la section 6.2 pour mettre à jour toutes les coordonnées d'affichages des différents médias dans la vue TimeLine. Comme cela on va avoir un affichage cohérent et à jour. Pour la mise à jour des autres vues, le mécanisme de synchronisation s'en chargera.

7.4.2 Suppression d'une contrainte

Sélection de la contrainte

Cette partie peut être effectuée facilement en repérant la dite contrainte dans la liste associée à la vue Time Line. La sélection de la contrainte entraîne implicitement une sélection des médias et des délais étant impliqués dans celle-ci.

Modification du graphe temporel

La modification du graphe temporel est assez complexe; nous allons énumerer succintement différents cas possible :

- Ajout de délai :
  On pourrait avoir un pareil cas si le noeud matérialisant une des extrémités des médias compris dans la contrainte ne contient aucun arc dans sa liste d'arcs entrants ou bien d'arcs sortants suivant le cas.
- Suppression de délai :
  Il est évident que les différents délais impliqués dans une contrainte vont disparaître avec la supression de celle-ci.
- Duplication de noeuds
  Cette partie est d'une très grande complexité. Pour bien la montrer, voici deux petits exemples de présentation plus ou moins semblables. Dans un cas il y aura une duplication de noeuds et pas dans l'autre, malgré un graphe de départ identique à première vue comme le montre la figure 7.4 et la figure 7.5.

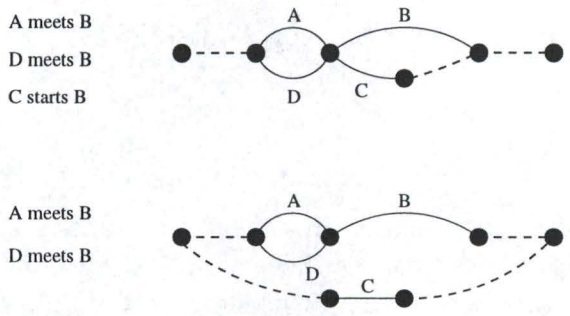

FIG. 7.4 – *Suppression d'une contrainte avec duplication de noeud*

Modifiation de la liste de contraintes

Il ne faut pas oublier de retirer la contrainte obsolète de la liste de contraintes, sinon la cohésion entre les différentes vues risque de ne plus être respectée.

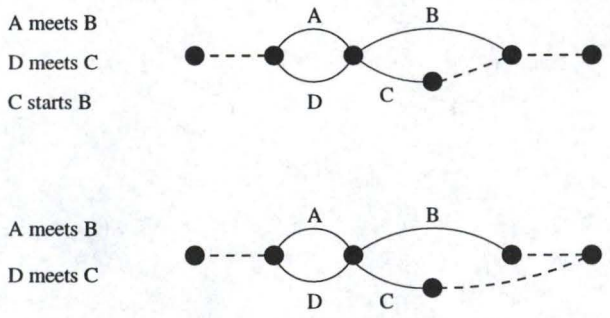

FIG. 7.5 – *Suppression d'une contrainte sans duplication de noeud*

Modification de l'affichage

Encore une fois, cette opération est très simple. Si besoin(si un ou plusieurs délais ont été ajoutés ou bien enlevés), il suffit de reparcourir le graphe pour metrre à jour les coordonéées des objets affichés dans la vue Time Line. Et ensuite de les présenter de nouveau à l'ecran.

7.5 Déplacement d'un objet dans le scenario temporel

Pour ce faire, il faut dans un premier temps éliminer toutes les contraintes de l'objet aussi bien les contraintes temporelles que les contraintes spatiales. Ces dernieres pourraient rester, mais la présentation des objets ne se produira pas en même temps. Si on les laisse, ceci pourrait cependant perturber l'auteur. En effet, le placement de certains médias serait inexpliqué rien qu'avec un simple regard sur l'affichage.

La seconde étape serait dans un premier temps de retirer l'objet et par la suite de le réinsérer à l'endroit voulu. On pourrait comparer cet enchaînement d'action à un couper/coller. En pratique, il faut éliminer l'objet du graphe le comprenant(Elimination de l'arc lui-même et des deux délais le reliant au noeud de départ et de fin de présentation). Puis le reinsérer à sa nouvelle place. Cela revient à effectuer une édition de la structure. Tous les principes énoncés à la section 4.2 sont aussi d'application ici. Si on veut seulement déplacer l'objet dans le même composite englobant, alors cette étape est inutile. Du moins, pour la généricité de l'implémentation, on peut tout de même l'effectuer.

La troisième étape est facultative. Le but ici est de "recoller" l'objet dans un nouveau jeu de contraintes pour affiner son nouvel instant de présentation. Pour ce faire, il suffit d'ajouter une ou plusieurs contrainte(s) à ce média par le mécanisme de sélection énoncé à la sous-section 7.4.1 à la page 84.

Chapitre 8

Conclusion

8.1 Synthèse

Comme nous l'avons vu tout au long de cet exposé, MADEUS est un éditeur de documents multimédia basé sur une approche relationnelle. Les différentes relations sont matérialisées par des contraintes. Cette approche permet à l'auteur de spécifier un scénario facilement sans se préoccuper de la spécification des instants de début et de fin des médias. En effet, le système analyse le jeu de contraintes temporelles insérées dans le document et fournit une solution particulière. Cette façon de procéder impose une phase de test de cohérence et de formatage assez lourde.

Dans cet éditeur, nous nous sommes concentrés sur la dimension temporelle des documents. Notre but principal étant l'édition directe dans une vue graphique, nous devions auparavant afficher le scénario temporel. Ceci impliquait un traçage à l'écran du graphe associé au document à présenter. Pour ce faire, nous avons opté pour un affichage prennant en compte des caractéristiques propres à l'affichage d'un scénario temporel. Voici les deux plus importantes :

- Nous préconisons un affichage horizontal des médias. Pratiquement ceci implique une duplication d'affichage pour certains noeuds du graphe temporel.
- Nous préconisons également un respect à l'affichage de la durée de l'objet associé un arc. Chaque rectangle représentant les différents composants du scénario doivent donc posséder une longueur proportionelle à la durée spécifiée pour l'objet représenté par celui-ci.

Dans la pratique, vu la durée du stage nous avons opté pour la solution présentée dans le chapitre 6. Dans une phase future du développement il n'est pas à exclure pour des raisons de performance ou bien pour des questions de meilleur agencement spatial que nous nous tournions vers une adaptation des algorithmes de dessin de graphe présentés dans le chapitre 5.

Le résultat obtenu avec cette méthode basique n'est pas parfait pour certaines

classes de doucments comme par exemple ceux découpés en beaucoups de scènes. Pour rémédier à ce petit inconvénient, nous avons donc mis sur pieds plusieurs améliorations pouvant être apportées à l'affichage original. Ces améliorations permettent entre une meilleure capacité d'édition, et ce sur deux tableaux :

- La méthode de limitation des croisements décrite à la section 6.5 engendre une meilleure lisibilité du scénario temporel. De ce fait, elle perment une meilleure perception des contraintes existantes.
- Les deux méthodes de "compactage" de l'affichage exposées aux sections 6.4 et 6.6 permettent une augmentation des informations disponibles sur la surface de travail. Dans le cas de gros document, l'utilisation fréquente des ascenceurs permettant d'accéder aux parties non visibles du scénario sera diminuée.

La possibilité offerte à l'auteur de passer d'une forme à l'autre d'affichage permet pour celui-ci une liberté de représentation suivant la nature de son document ou bien suivant la complexité de la manoeuvre d'édition à effectuer.

Du point de vue édition de la durée d'un média, la solution proposée à la section 4.4 permet un retaillage limité par les bornes de l'intervalle de durée de l'objet à modifier. Cette limitation améliore les performances du système en n'invoquant pas un recalcul de la cohérence temporelle. De plus, avec cette limitation les nouvelles données assignées sont toujours valides. Ce qui ne se produirait pas souvent dans le cas contraire. Un nombre important de refus de modifications pourrait irriter la personne utilisant le logiciel.

Pour terminer, il ne faut pas oublier le problème du passage à l'échelle lors du traîtement de gros document. Le compactage de l'affichage est un maigre début de solution. Si on remarque bien, tous les systèmes auteurs rencontre des difficultés pour faire face à ce problème. Seul iSHELL innove en permettant l'ouverture d'une composante de la hiérarchie dans une fenêtre annexe. C'est peut être dans cette direction que nous trouverons par la suite une façon efficace de régler ce problème.

8.2 Perspectives

Voici une petite liste des perspectives à ce travail permettant d'améliorer l'outil du point de vue visualisation et édition d'informations :

- La suite logique de ce travail sera de continuer l'édition directe dans la vue avec les possibilités d'ajout et de suppression de contraintes entre deux objets. Cette opération est assez complexe et mérite une grande attention comme nous avons pu le remarquer à la section 7.4. Pour une édition parfaitement intuitive, c'est la composante essentielle à développer en premier lieu.

- Les différentes méthodes d'amélioration de l'affichage sont automatiques. C'est le programme qui opère les changements de coordonées sans interaction de l'utilisateur. Or deux personnes différentes ne se représentent pas de la même façon le scénario temporel d'un même document. Ils placeraient spatialement les objets à des endroits différents. C'est pourquoi il serait intéressant de permettre une réorganisation de l'affichage manuel dans la vue Time Line. Rappelons qu'une brève idée de solution a été présentée à la section 7.2.
- Il ne faut pas oublier de traiter la dimension sémantique des documents en permettant une visualisation des hyperliens dans la vue Time Line.
- Cette dernière perspective n'apportera rien d'extraordinaire au point de vue des fonctionnalités de l'éditeur, mais elle n'est quand même pas à négliger. La représentation des différents types de médias dans la vue Time Line à été créée sans aucun but artistique, en effet, la durée du stage ne nous a pas permis de nous attarder sur l'apparence externe de la vue. Pourtant celle-ci est primordiale pour d'une part bien différencier les types de média et d'autre part pour accrocher un auteur qui aime utiliser un outil joli au regard et convivial.

Bibliographie

[1] Allen J.F. "Maintaining Knowledge about Temporal Intervals", *in Communication of the ACM*,Vol. 26 (No. 11), pp.832-843,november 1983

[2] Authoring Languages,
URL : `http://www.mcli.dist.maricopa.edu/authoring0/lang.html`

[3] Bes F. "Spécification hiérarchique de scénario temporel à base de contraintes", Rapport de DEA Informatique de l'Université Joseph Fourier (Grenoble 1), Juin 1998.

[4] Carcone L. "Formatage spatial dans un environnement d'édition / présentation de documents multimédia", Mémoire, Cnam, décembre 1997

[5] Chisogne C. "Systèmes de contraintes temporelles appliqués aux documents multimédia structurés", mémoire de maîtrise informatique FUNDP, à paraître septembre 1999

[6] daVinci, URL : `http://www.tzi.de/~davinci/`

[7] Di Battista G., Eades P., Tamassia R and Tollis I.G. "Algorithms for Drawing Graphs : an Annotated Bibliography", *in Computational Geometry : Theory and Applications*, no. 4, pp. 235-282, 1994

[8] DIRECTOR,
URL : `http://www/macromedia.com`

[9] Fary I. "On Straight Lines Représentation of planar graphs", *in Acta Sci. Math. Szeged*, vol 11, pp 229-233, 1948

[10] F. Cruz I. and Tamassia R. "Graph Drawing Tutorial",
URL : `http://www.cs.brown.edu/people/rt/papers/gd-tutorial/gd-constraints.pdf`

[11] Feng Q. "Algorithms for Drawing Clustered Graphs", thesis of the Departement of Computer Science and Software Engeneering, The University of Newcastle, april 1997

[12] Fichefet J. "Introduction à la théorie des graphes"

[13] Goffinet L."Le multimédia et ses systèmes auteurs", FUNDP Namur, URL: `http://www.info.fundp.ac.be/~lgo/Authoring/systemes.html`

[14] GRiNS tutorial guide,
URL :`http://www.oratrix.com/downloaddir/tutorials.zip`

[15] GRiNS quick start guide,
URL: `http://www.oratrix.com/downloaddir/quickstart.pdf`

[16] Hyperbolic Tree SDK,
URL: `http://www.inxight.com/Products/Developer/Hyperbolic_Tree.html`

[17] iSHELL Download
URL: `http://www.tribeworks.com/Public/Content/Application.html`

[18] iSHELL Online Documentation,
URL: `http://www.tribeworks.com/Help/index.htm`

[19] Jourdan M. Layaïlda N. and Roisin C. "A survey on authoring techniques for temporal scenarios of multimedia documents",*in Vol. Handbook of Internet and Multimedia Systems and Applications*, part 1, Tools and Standarts, CRC Press, April 1998

[20] Jourdan M., Roisin C., Tardif L. "A Scalable Toolkit for Designing Multimedia Authoring Environments", numéro spécial 'Multimedia Authoring and Presentation: Strategies, Tools, and Experiences' de Multimedia Tools and Applications Journal, Kluwer Academic Publishers, à paraître 1999.

[21] Layaïlda N. "Madeus: système d'édition et de présentation de documents structurés multimédia", Doctorat Informatique, UJF Grenoble, juin 1997

[22] Mackinlay J. D., Robertson G. G. and S. K. Card. "The perspective wall: Detail and context smoothly integrated." ACM Conference on Human Factors in Computing Systems (CHI '91), ACM: 173-179, 1991

[23] Mutzel P. and Gutwenger C. "AGD, Algoritms for Graph Drawings: user manual",
URL: `http://www.mpi-sb.mpg.de/AGD/online_manual.html`

[24] Sander G. "Graph Layout through the VGC Tool", technical report A03/94, Universität des Saarlandes, FB 14 Informatik, 1994

[25] Sander G. "Visualization of Compiler Graphs: User Documentation V.1.30", 1995

[26] Siglar J. "Mulimedia Authoring Systems FAQ Version 2.23",
URL: `http://www.tiac.net/users/jasiglar/MMASFAQ.HTML`, 4 avril 1999

[27] Simone L. MACRIOMEDIA DIRECTOR 7, *in PC MAGAZINE first look*
URL: `http://www.zdnet.com/products/stories/reviews/0,4161,2188242,00.html`, 15 janvier 1999

[28] Sugiyama K., Tawaga S. and Tada M., "Methods for Visual Understanding of Hierarchical Systems", *in IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 21, no. 2, pp. 144-164, 1985

[29] Tamassia R., Di Battista G. and Batini C. "Authomatic graph drawing and Readability of Diagrams", *in IEEE Transaction on Systems, Man and Cybernetics*, ol SMC-18, no. 1, pp 61-79, 1988

[30] Tamassia R. "Advances in the Theory and Practice of Graph Drawing", Departement of Comuter science Brown University, URL: `http://www.cs.brown.edu/poeple/rt/papres/ordal96/ordal96.html`

[31] Tardif L. "Visualisation du scénario temporel d'un document multimédia", rapport DEA IMAG Grenoble, juin 1997

[32] Tarjan R.E. "Algorithm Design", *in Communication ACM*, Vol. 30, no. 3, pp 205-212, 1987

[33] Tutte W.T. "How to Draw a graph", *in Proc. London Math Soc., Vol. 3 no. 13*, pp 743-768, 1963

[34] W3C. "Syncrhonized Multimedia Integration Language (SMIL) 1.0 Specification", URL :`http://www.w3c.org/TR/REC-smil-19980615`

[35] W3C Working Draft "Synchronized Multimedia Integration Language (SMIL) Boston Specification" , URL: `http://www.w3.org/TR/smil-boston/`, 20 August 1999

[36] Yu J. and Xiang Y. "Hypermedia Presentation and Authoring System.", *proceedings of the sixth international World Wilde Web*, pp 153-164, Santa Clara, USA, 7-11 Avril 1997

[37] Yu J. "A simple, intuitive hypermedia synchronization model and its realization in the browser/Java environment." Technical Note 1997-027a, Digital Equipment Corporation Systems Research Center, Palo Alto, CA, April 1998.