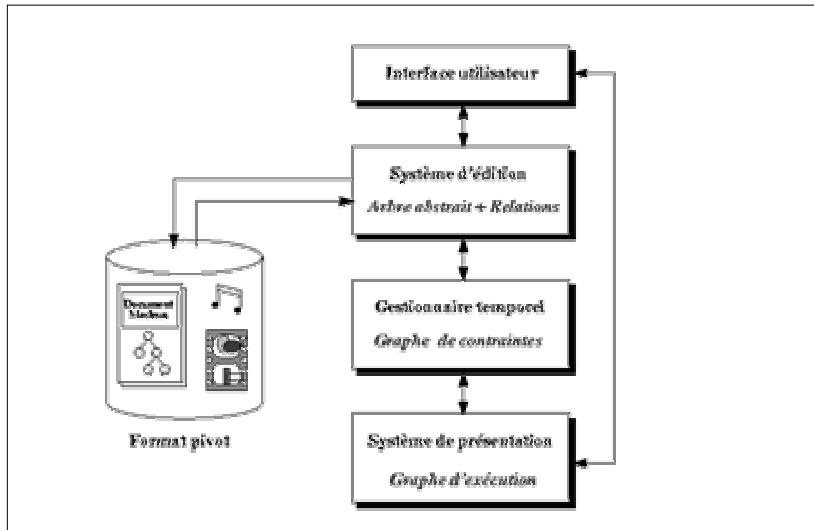
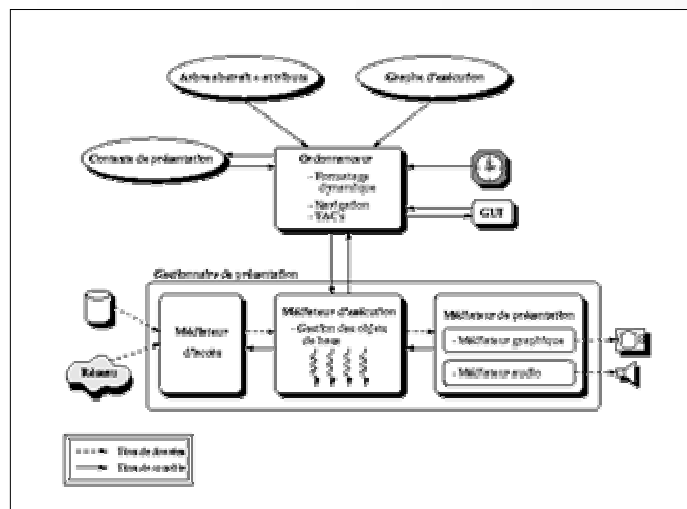


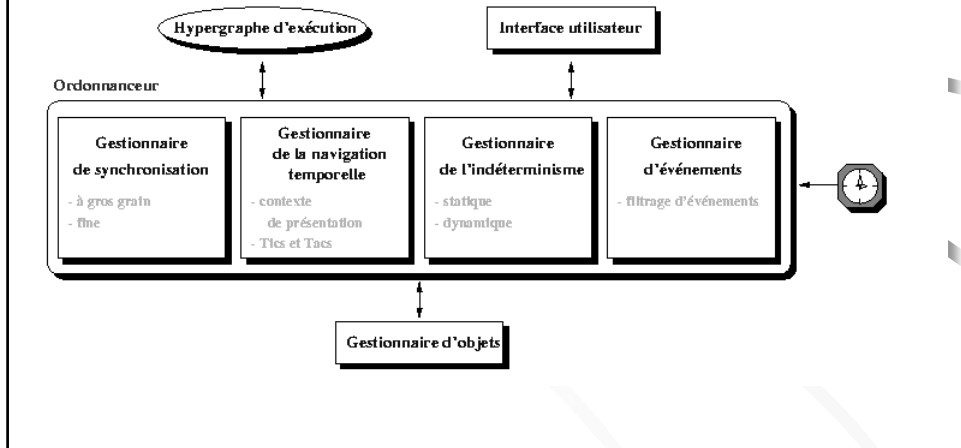
Architecture générale d'un SDMM



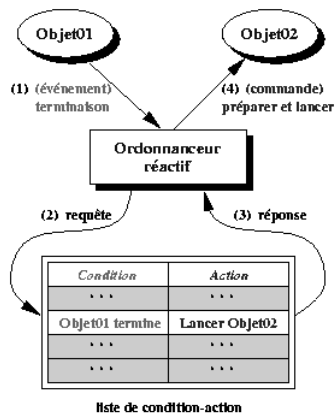
Architecture d'un système de présentation



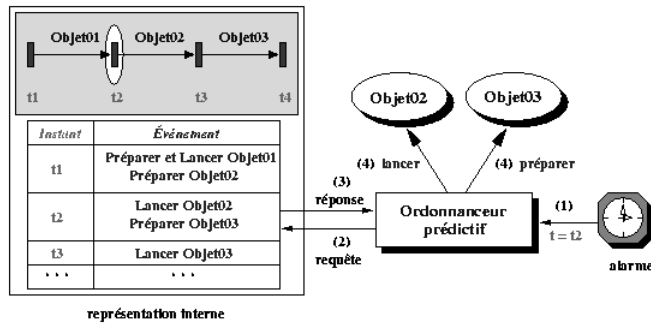
.. Plus précis



Ordonnancement : L'approche réactive

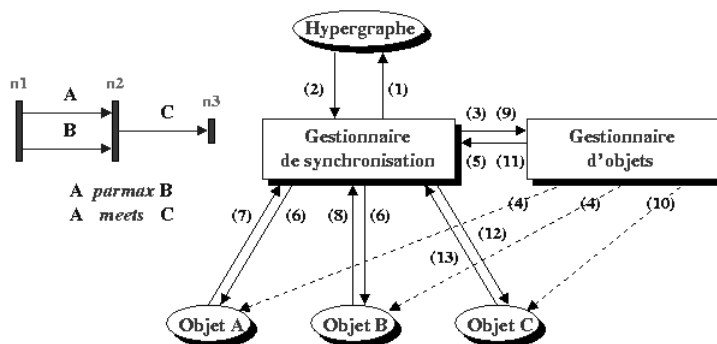


L'approche prédictive

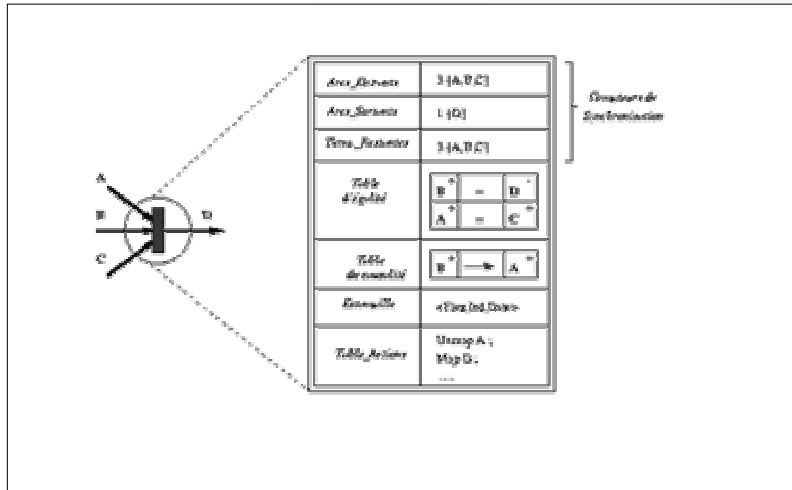


La réalite : meilleure approche hybride

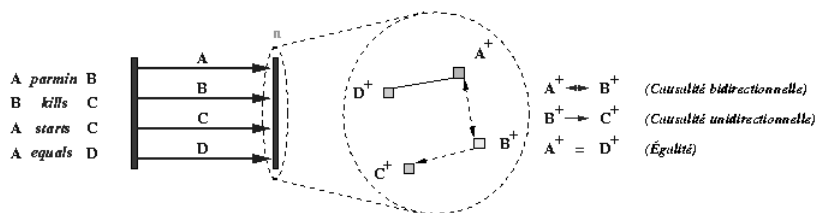
Synchronisation inter-objets



Synchronisation inter-objets

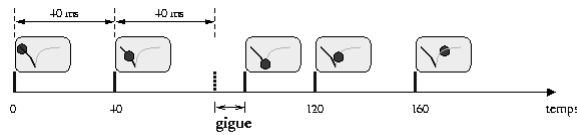


Synchronisation inter-objets

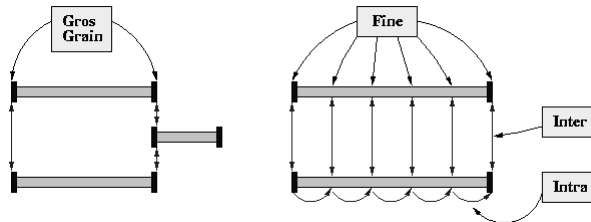


Assurer la synchronisation

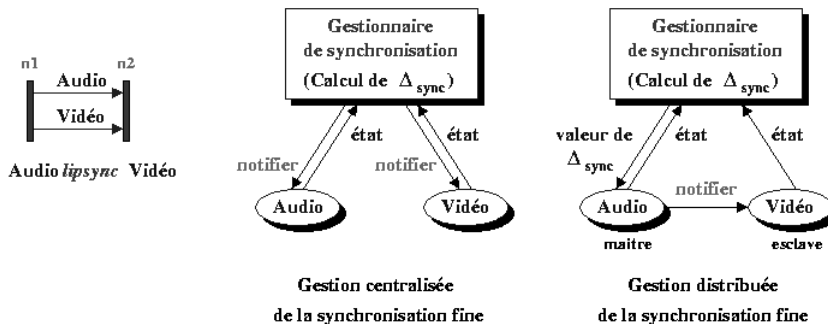
- Synchronisation intra-objets :



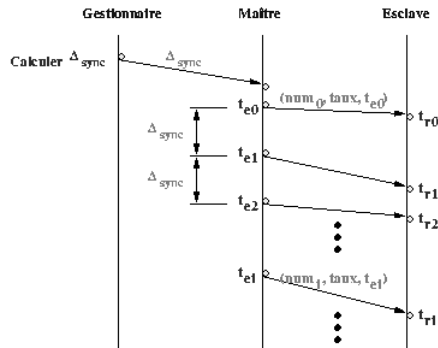
- Synchronisation inter-objets (Gros grain) :
- Synchronisation lip-sync (Fine) :



Assurer la synchronisation



.... suite



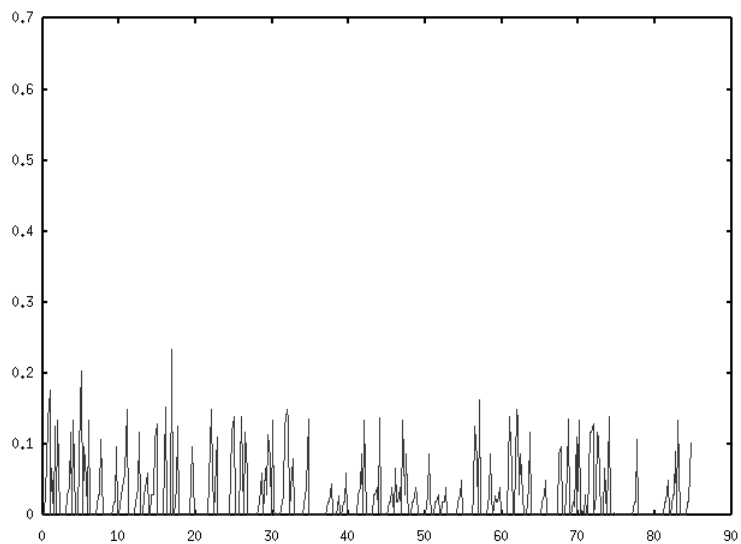
Jouer_Video_avec_intra-synchronisation() {

```

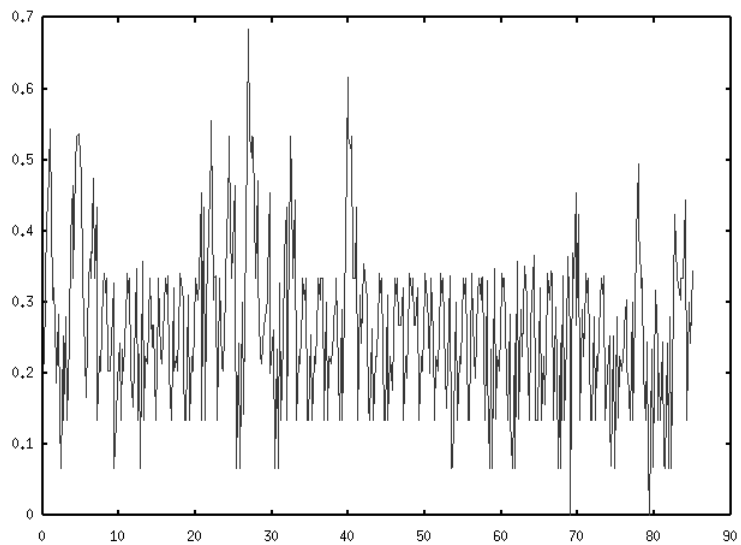
1. /* Initialiser le compteur d'images pour que tpres
2. définisse la date limite de présentation de l'image
3. courante */
4. numéro_image = 1
5. /* Boucle de décompression et de présentation */
6. Pour chaque image in vidéo {
7. /* Calculer la date de la présentation de l'image
8. en fonction du numéro de l'image et le taux de
9. présentation de la vidéo */
10. tpres = tdébut + numéro_image/taux_de_présentation
11. /* Calculer la valeur de gigue */
12. gigue = tactuel - tpres
13. Si (gigue < Deltagigue) {
14. Décompresser_Image ()
15. Afficher_Image ()
16. /* Calculer le temps qui reste jusqu'à
17. l'instant tpres */
18. gigue = tactuel - tpres
19. Si (gigue < 0) {
20. /* Bloquer le processus pour un délai égal à
21. ((-1) * gigue) unité de temps, c'est-à-dire
22. jusqu'à la date de la présentation tpres */
23. Bloquer_Processus ((-1) * gigue)
24. }
25. } Sinon {
26. /* Il est trop tard : Sauter cette image */
27. }
28. numéro_image = numéro_image + 1
29. }}

```

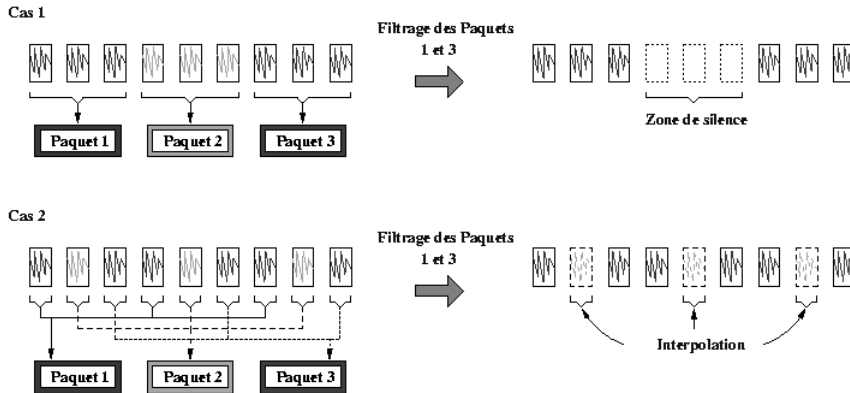
Gigue observée pour $\Delta = 0.1$ s CCPU=3



Gigue observée pour $\Delta = 0.1$ s CCPU=5



Filtrage de données : principe



Filtrage de données Mpeg

- Le format Mpeg tire profit des similarités entre les images d'une séquence vidéo, par prédiction et interpolation, afin de réduire la redondance de l'information contenue dans ces séquences.
- La compression crée ainsi de nouvelles dépendances entre les différentes unités de présentation liées uniquement à leur encodage.

Filtrage de données Mpeg

La compression produit trois différents types d'images :

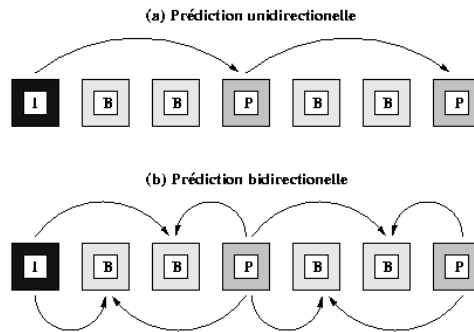
- I (*Intra frames*) : Elle est compressée en utilisant seulement les informations existantes dans l'image lui-même. Ce type d'images représente des points d'accès potentiels dans le flot vidéo compressé.
- P (*Predicted frames*) : Elle est compressée en utilisant comme référence la précédente plus proche image de type I ou P. Ce type de codage est appelé *prédiction unidirectionnelle*. Comme l'image de type I, une image de type P peut servir comme référence de prédiction pour les images de type P et B.
- B (*Bidirectional frames*) : Dans sa compression, elle utilise deux images comme références de prédiction : la précédente et la suivante plus proches images de type I et/ou P. Ce type de codage est appelé *prédiction bidirectionnelle* ou *interpolation*.

Filtrage de données Mpeg

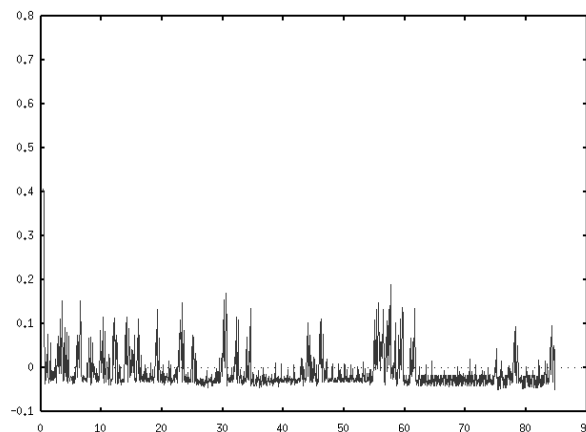
- Celles-ci sont très importantes à considérer pour appliquer la technique du saut d'images.
- Dans, MPEG le contenu de certaines images (B ou P dans le codage) ne peut être reconstitué que si leur image de référence (images de type I ou P dans le codage) a été préalablement traitée.
- La succession des images I, P, et B pour une MPEG est définie par un motif ou *pattern* comme *IBBPBBPBBPBB...*

Filtrage de données Mpeg

- Pour filtrer une image MPEG, il faut donc garantir que toutes les images qui en dépendent sont également sautées. La technique de saut d'images commence par sauter d'abord les images de type B, puis le type P et enfin le type I.



Influence du filtrage Mpeg sur la gigue

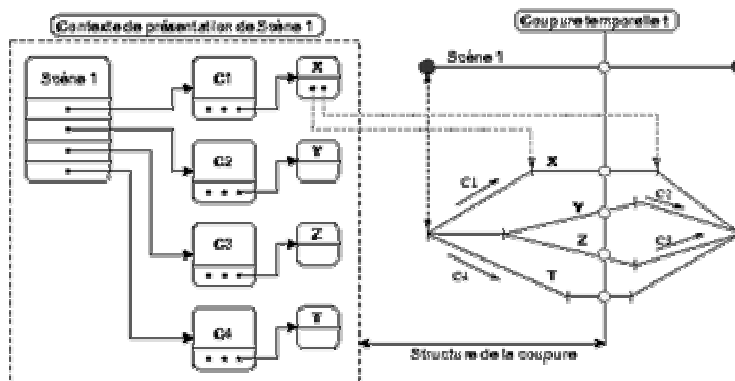


La gigue d'une vidéo utilisant le saut d'images (utilisation du cpu = 100%, charge du cpu = 3)

Gestionnaire de la navigation temporelle

- Notions fondamentales :
 - Ancre de départ
 - Ancre d'arrivée
- Types de navigation
 - Navigation intra- inter-documents
 - Navigation pas à pas

Contexte de présentation



Navigation temporelle

- Le contexte courant de présentation d'un scénario S à l'instant *actuel* est une liste dont les items sont les objets de base actuellement dans l'état actif avec leurs valeurs *offset*.
- Le contexte courant est défini par la fonction CP_{actuel} comme suit :

$CP_{actuel}(S) = \{ (Objet, Offset(Objet)) / (Objet \text{ in } S) \}$
et $(Nature(Objet) = BASIC)$
et $(Date_effective_de_début(Objet) \leq temps_actuel)$
et $(Date_nominale_de_fin(Objet) > temps_actuel)$
et $(Offset(Objet) = temps_actuel - Date_eff_de_début(Objet))$
}

Navigation temporelle

- L'Ordonnanceur utilise la solution temporelle calculée par le formateur statique pour déterminer, si besoin, le contexte de présentation CP_t d'un instant t . Ainsi, pour évaluer CP_t , il faut utiliser :
- les informations de la structuration logique et hiérarchique du document maintenues dans l'hypergraphe,
- les informations temporelles des dates de début et de fin de chaque objet calculées par le formateur statique (comme $Date_nominale_de_début$, etc.),
- les informations de l'état (actif ou non) et de la nature des objets (BASIC ou COMPOSITE).
- À partir de ces informations, le temps écoulé de la présentation d'un objet à un instant t peut être calculé dynamiquement par l'ordonnanceur

Navigation temporelle

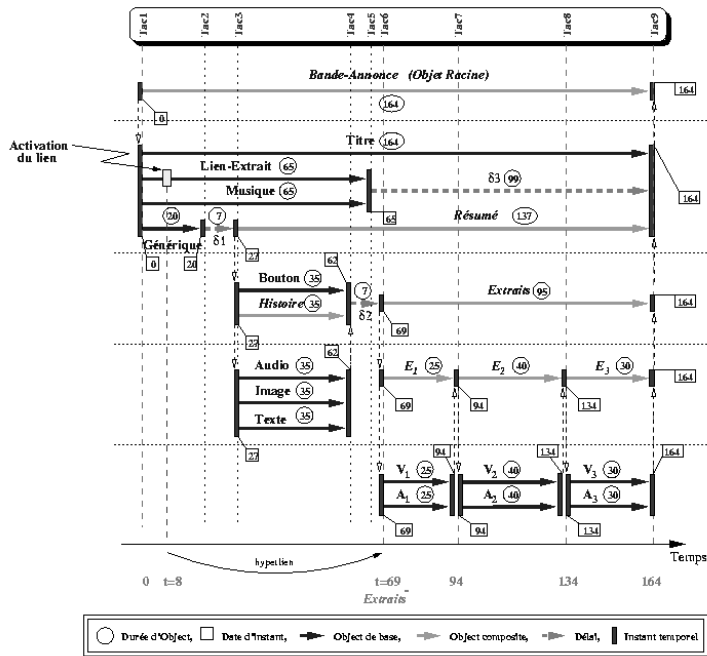
On peut de même calculer le contexte de présentation de tout instant t en utilisant la date nominale du début des objets, calculée par le formateur statique, au lieu de la date effective, comme suit :

$$\begin{aligned} \text{CP}_t(S) = \{ & (\text{Objet}, \text{Offset}(\text{Objet})) / (\text{Objet in } S) \\ & \text{et } (\text{Nature}(\text{Objet}) = \text{BASIC}) \\ & \text{et } (\text{Date_nominale_de_début}(\text{Objet}) \leq t) \\ & \text{et } (\text{Date_nominale_de_fin}(\text{Objet}) > t) \\ & \text{et } (\text{Offset}(\text{Objet}) = t - \text{Date_nominale_de_début}(\text{Objet})) \\ & \} \end{aligned}$$

Mise en oeuvre des opérations de navigation

Quand cet hyperlien est activé à un instant t , l'opération de navigation est effectuée en passant par les étapes suivantes :

1. *sauvegarder* le contexte actuel de présentation (CP_t) dans la pile du contexte de présentation ;
2. *stopper* la présentation, et libérer ensuite les ressources allouées ;
3. *calculer* le contexte de présentation correspondant à l'instant de début de B (CP_B) en utilisant les informations de l'hypergraphe ;
4. *mettre à jour* le contexte actuel de présentation en lui affectant le contexte calculé CP_B puis, reprendre la présentation. La présentation commence à partir de l'instant de début de B ;
5. dès que la présentation de la destination de l'hyperlien est terminée, le contexte CP_t peut être *restauré* à partir de la pile et utilisé comme le contexte courant de présentation. Par conséquent, la présentation peut être reprise à partir de l'instant t .

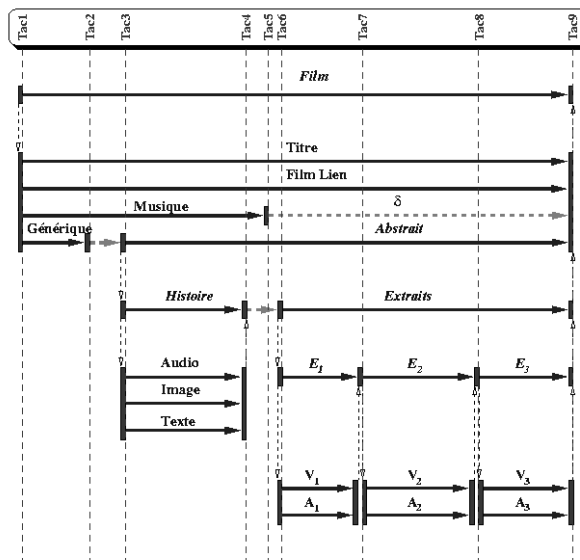


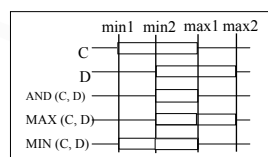
...

- L'activation de l'hyperlien à l'instant $t=8$.
 - Le contexte actuel de présentation ($CP_{t=8}$) est :
 - $CP_{t=8} = \{(\text{Titre}, 8), (\text{Lien-extrait}, 8), (\text{Musique}, 8), (\text{Générique}, 8)\}$,
- et le contexte de présentation de l'ancre destination ($CP_{\text{Extraits}} \equiv CP_{t=69}$) est :
- $CP_{\text{Extraits}} = \{(\text{Titre}, 69), (V_1, 0), (A_1, 0)\}$

Navigation pas à pas

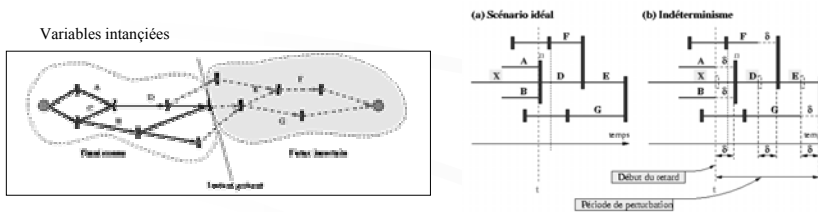
- $CP_{Tac1} = \{(Titre, 0), (Lien-extrait, 0), (Musique, 0), (Générique, 0)\}$,
- $CP_{Tac2} = \{(Titre, 20), (Lien-extrait, 20), (Musique, 20), (\delta 1, 0)\}$,
- $CP_{Tac3} = \{(Titre, 27), (Lien-extrait, 27), (Musique, 27), (Audio, 0), (Image, 0), (Texte, 0)\}$,
- $CP_{Tac4} = \{(Titre, 62), (Lien-extrait, 62), (Musique, 62), (\delta 2, 0)\}$,
- $CP_{Tac5} = \{(Titre, 65), (\delta 2, 3)\}$,
- $CP_{Tac6} = \{(Titre, 69), (V_1, 0), (A_1, 0)\}$,
- $CP_{Tac7} = \{(Titre, 94), (V_2, 0), (A_2, 0)\}$,
- $CP_{Tac8} = \{(Titre, 134), (V_3, 0), (A_3, 0)\}$,
- $CP_{Tac9} = \text{emptyset}$ (l'instant de la fin de la présentation).





Formatage dynamique : principe

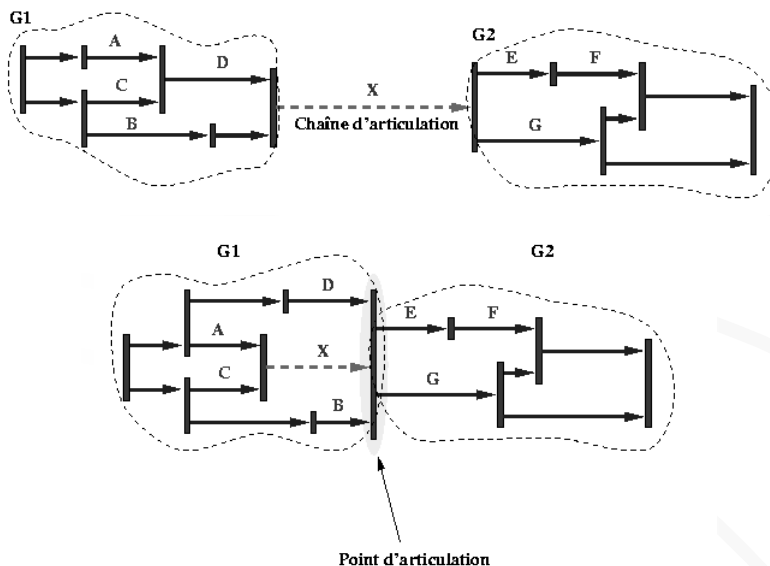
- Ordonnancement dans un environnement à durées incertaines
- Résolution et maintien de solution dynamique (Passé, Présent, Futur)
- **Approche** = Partir d'une solution et la remettre en cause en cours d'exécution



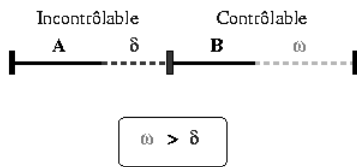
Difficulté :

- Résolution se fait en temps réel (au rythme des observations)
- Irréversibilité du temps (pas de remise en cause du passé!)
- Pour aboutir à un succès : $P(\text{cohérence}) \forall v_{i1}, \dots, v_{in} \exists v_{c1}, \dots, v_{cn}$

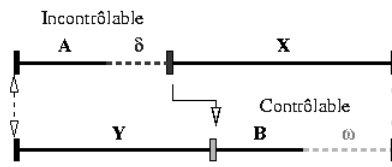
Cas faciles



Plus complexe



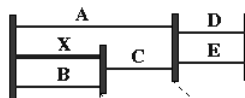
(a)



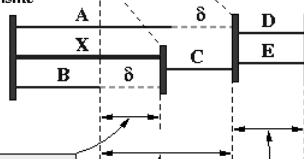
(b)

Encore plus

(a) Scénario idéal



(b) Effet d'indéterminisme

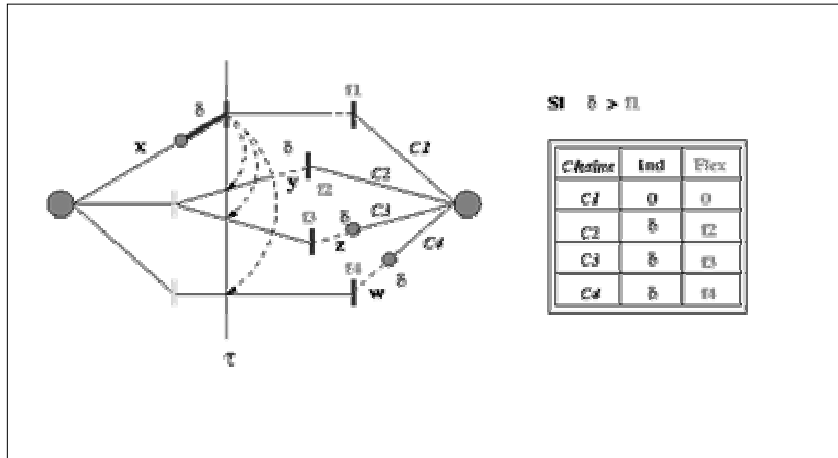


Δ_{ind} : Période d'indéterminisme (l'état indéterministe)

Période de perturbation

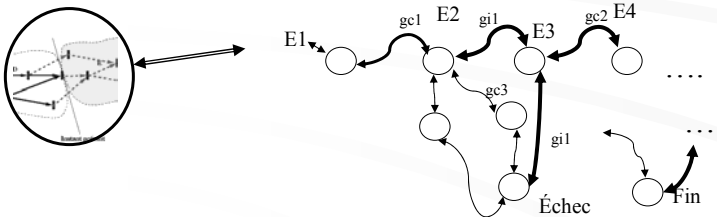
Désynchronisation terminée

Utilisation de la coupure pour le formatage : a posteriori



Formatage dynamique : a priori

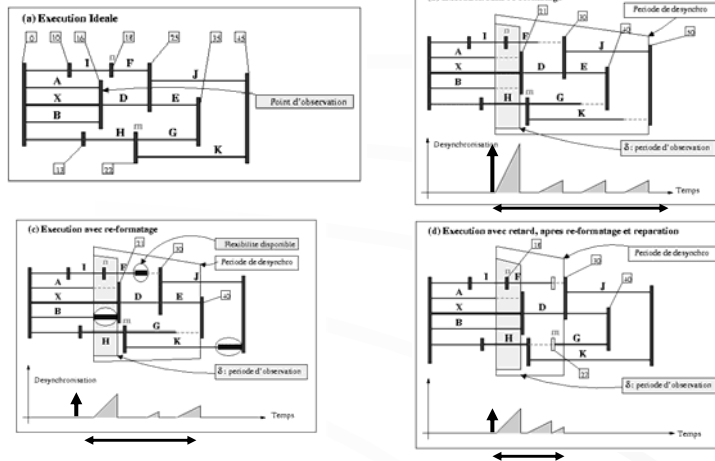
- Automate temporisé avec date d'échéances
- Une fonction d'étiquetage h : des éléments de \rightarrow
 - Gardes (sommets) sont de deux types (contrôlables, incontrôlables)
 - Résolution = partie de jeu entre l'ordonnanceur et l'environnement



Approche

- Rechercher localement un chemin qui permet d'éviter l'état d'échec.
- L'état d'échec est "récupérable" mais il faut éviter le blocage du scénario.

Reprise dans le cas d'un échec



Plan du cours

- Système multimédia adaptables
- Documents multimédia
- Système multimédia et modélisation
- Edition et présentation de documents multimédia adaptables
- Profiles et négociation
- Systèmes de présentation