

Slide 1



Cours No 9 - Médiation de requêtes



Intégration d'informations

L'intégration d'informations est un besoin majeur dans la gestion de contenus Web.

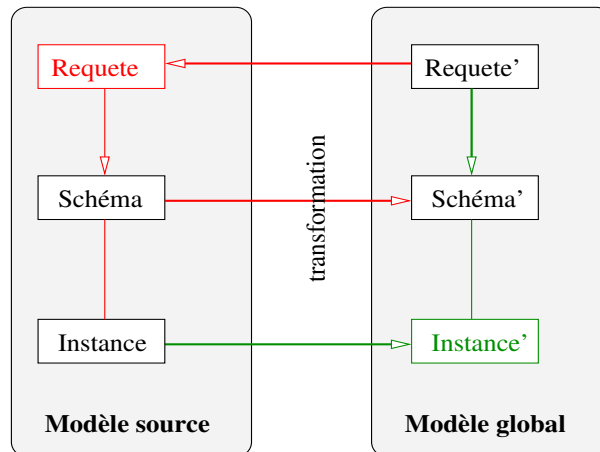
Slide 2

- Problème : sources d'informations
 - hétérogènes : systèmes, interfaces, modèles, formats, schémas, sémantique
 - réparties
 - autonomes : gestion indépendante, pas de modification d'applications existantes
- **Objectif : Offrir un point d'accès unique et homogène**



Migrer les requêtes ou les données ?

Slide 3



Approche Médiateur : accès directe

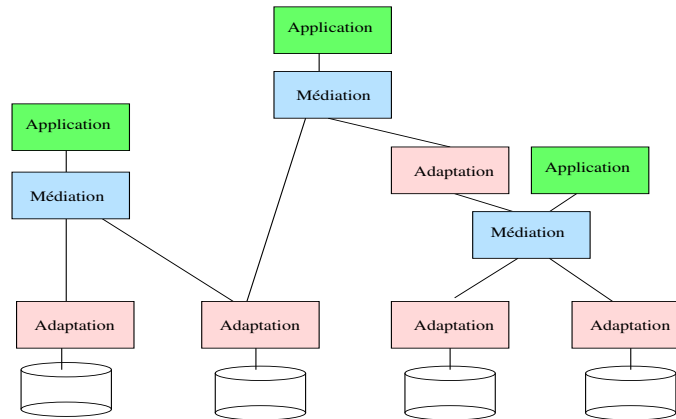
Slide 4

- Approche "paresseuse", pas de matérialisation a-priori des données
- Migration de requêtes vers les sources
- Avantages : cohérence (données d'origine)
- Inconvénients : performance, traduction de requêtes, dépendance des capacités de sources



La séparation des tâches

Slide 5



(Architecture de référence DARPA I3)



Adaptateur et Médiateur

Slide 6

- Médiateur :
 - décompose les requêtes globales en plusieurs requêtes sources
 - intègre les résultats obtenus par les sources
- Adaptateur (wrapper) :
 - évalue les requêtes locales envoyées à une sources

Le médiateur doit prendre en compte les capacités et le schéma exporté par chaque adaptateur.

Slide 7

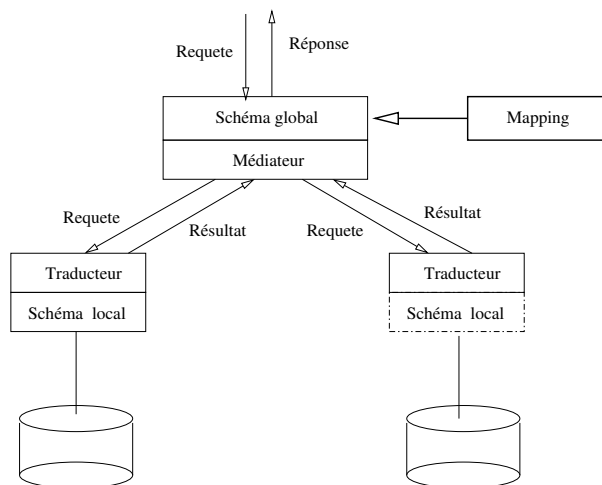


Description des sources



Architecture de Médiation

Slide 8





Comment intégrer une source dans un médiateur?

Slide 9

- Il faut créer des liens ou “mappings” entre le schéma local de la source et le schéma global.
- Ces mappings servent à traduire une requête utilisateur en une ou plusieurs requêtes sources.

Intégration = définition de vues



Définition de mapping

Slide 10

Le médiateur est une vue globale sur l'ensemble des sources (Global as View - GAV) :

- Avantage : simplicité de la traduction de la requête : algorithme traditionnel de remplacement de la vue par sa définition.
- Inconvénient : l'insertion, la suppression et la modification d'une sources demande un changement global du mapping.

Chaque source est une vue locale sur le médiateur (Local as View - LAV) :

- Avantage : chaque source est intégrée indépendamment de l'autre → flexibilité
- Inconvénient : complexité du problème de “réécriture de requêtes avec vues”



Exemples de médiateurs

Slide 11

Source	Interne	Externe	Approche	Exemple
rel.	rel.	semi-struct.	GAV	Tsimmis
rel.	rel.	logique de desc.	LAV	PicseI
XML	rel.	XML	LAV	Agora
XML	XML	XML	LAV/GAV	Xyleme
XML	XML	XML	GAV	e-XMLMedia
XML	XML	conceptuel	LAV	StyX

Slide 12



Trois exemples de médiateurs



e-XMLMedia

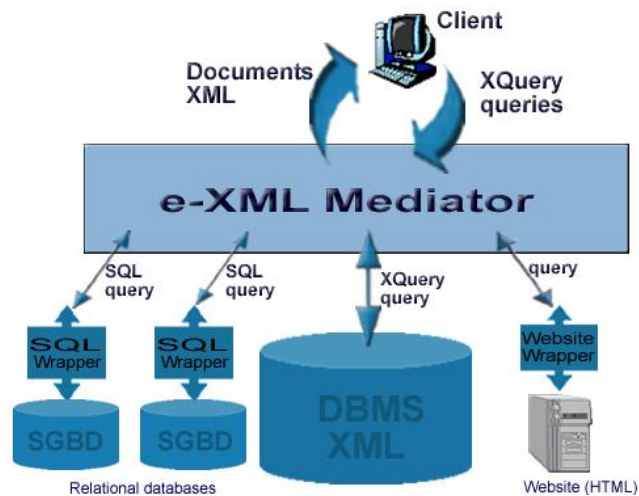
Slide 13

- Entrepôt XML : stockage de documents XML dans une base de données relationnelle; interrogation XQuery et SQL; XML/DBC
- Médiateur : vue XML sur sources hétérogènes; interrogation avec XQuery
- XMLizer : exportation/importation de documents XML dans une BD relationnelle
- XFE (XFORMS Engine) : gestion de formulaires décrits dans le standard XFORMS



Médiateur e-XMLMedia

Slide 14





Architecture du Médiateur

Slide 15

- Console d'administration : configuration du médiateur, enregistrement de nouveaux adaptateurs
- Gestionnaire de métadonnées : mappings schémas XML ↔ schémas relationnelles
- Analyseur/décomposeur ; traduction des requêtes, élaboration de plans d'exécution distribués, optimisation
- Évaluateur : délégation de sous-requêtes aux adaptateurs, collection des résultats, évaluation de la partie global du plan (agrégats, jointures) en mémoire, reconstruction du résultat
- Gestionnaire des connexions avec les adaptateurs



Xyleme

Slide 16

- Entrepôt XML : les documents sont stockés sans transformation
- Interrogation : l'utilisateur interroge les document à travers une description abstraite des documents.

Approche hybride d'intégration :

- intégration virtuelle de données matérialisées



Xyleme : Traitements sémantiques

Classification : Les documents XML sont organisés suivant une *hiérarchie prédéfinie de clusters* :

- Tous les documents d'une DTD appartiennent au même cluster
- Tous les documents d'un cluster sont stockés ensemble

Slide 17

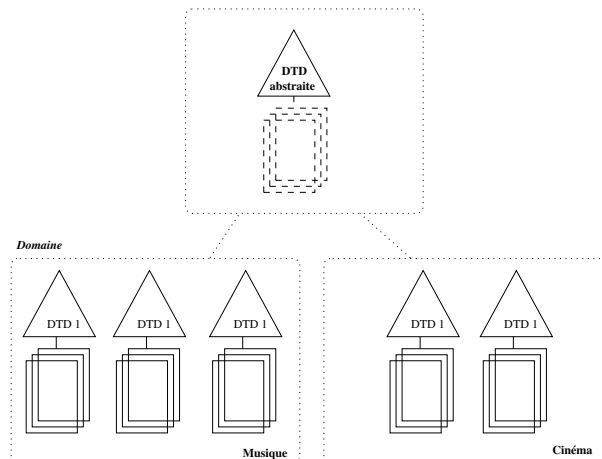
Intégration :

- Domaine = ensemble de clusters
- Les domaines ne sont pas forcément disjoints
- La structure d'un domaine est décrite par un *DTD abstraite*



Xyleme : DTD Abstraites

Slide 18





DTD abstraite

Fonctions :

Slide 19

- Définition d'un domaine sémantique
- Description de clusters de DTD concrètes
- Interface d'interrogation unique pour documents hétérogènes
- Schéma de structuration du résultat

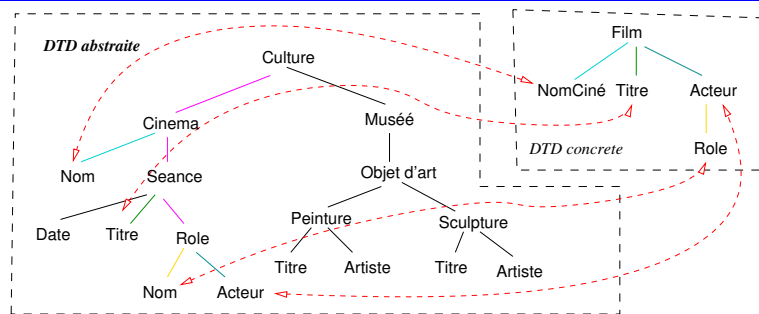
Structure :

- Arbre de concepts



Culture et Films

Slide 20



- m1: Culture/Cinéma/Nom ⇔ Film/NomCiné
 m2: Culture/Cinéma/Séance/Titre ⇔ Film/Titre
 m3: Culture/Cinéma/Séance/Role/Nom ⇔ Film/Acteur/Role
 m4: Culture/Cinéma/Séance/Role/Acteur ⇔ Film/Acteur



Interrogation

Langage de requêtes

Slide 21

- extension OQL + chemins + “contains”
- jointures (valeurs, liens)
- transformation de documents
- absence d'ordre

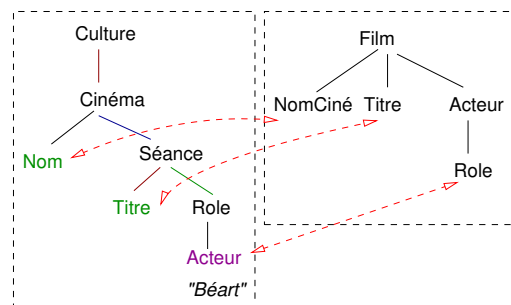


Traduction de Requête

```
SELECT cin/Nom, sea/Titre
FROM doc in Culture,
      cin in doc/Cinéma,
      sea in cin/Séance
WHERE sea/Role/Acteur = "Béart"
```

```
SELECT fil/NomCiné, fil/Titre
FROM doc in Films,
      fil in doc/Film
WHERE fil/Acteur contains "Béart"
```

Slide 22





Évaluation

Plan d'exécution :

- opération de base : *PatternScan*
- génération de plan d'exécution : union, jointures (sur les liens)

Évaluation :

- Machine interface :
 - génération du plan d'exécution
 - intégration des résultats
- Machine index :
 - contient le mapping pour un cluster
 - évalue des *PatternScan*

Objectif : Performance

Slide 23



StyX

Intégration de documents XML :

- Sources : capables d'évaluer des requêtes XQuery ou XPath
- Modèle global : graphe de concepts
- Règles de mapping :
expressions XPath ← chemins conceptuels

Slide 24



Architecture

Slide 25

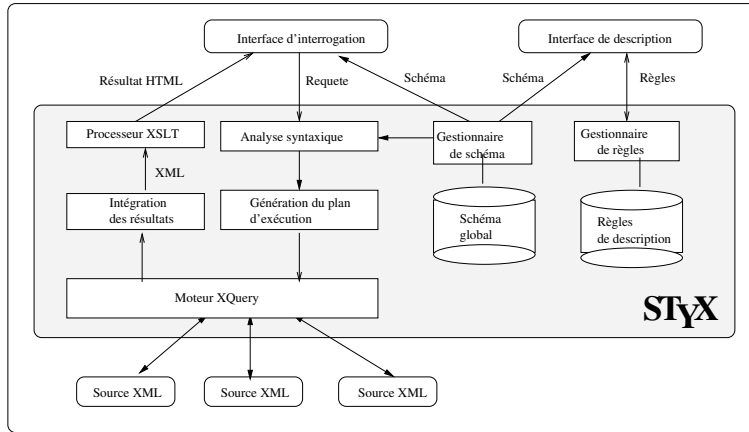
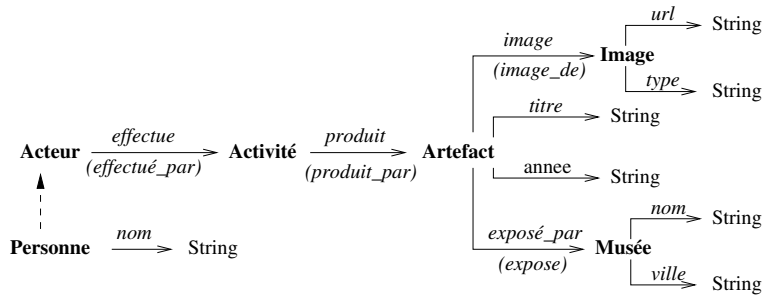


Schéma global : Graphe de Concepts et de Rôles

Slide 26





Règles de Mapping

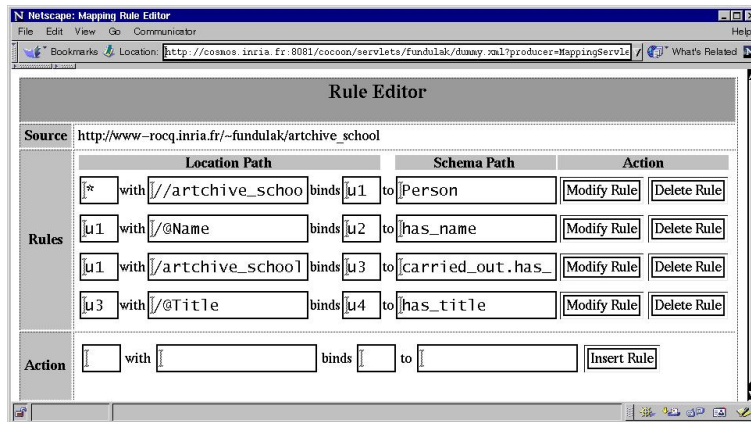
Slide 27

R_1 : `http://www.paintings.com/Peintre` as u_1 ← Personne
 R_2 : u_1 /Nom as u_2 ← nom
 R_3 : u_1 /Peintures/Peinture as u_3 ← effectue.produit
 R_4 : u_3 /@titre as u_4 ← titre
 R_5 : u_3 /@année as u_5 ← produit.année



Interface de description

Slide 28





Requête

Requête utilisateur :

Q_1 : **select** c
from **Personne** a , a .**nom** b , a .**effectue.produit.titre** c
where $b = \text{"Van Gogh"}$

Slide 29

Mapping :

R_1 : `http://www.paintings.com/Peintre` **as** u_1 ← **Personne**
 R_2 : u_1 /Nom **as** u_2 ← **nom**
 R_3 : u_1 /Peintures/Peinture **as** u_3 ← **effectue.produit**
 R_4 : u_3 /@titre **as** u_4 ← **titre**
 R_5 : u_3 /@année **as** u_5 ← **produit.année**



Traduction de Requêtes

Binding: $\{a \rightarrow R_1, b \rightarrow R_2, c \rightarrow R_3.R_4\}$

Traduction : appliquer le binding

$Q_1(a)$: **select** c
from `http://www.paintings.com/Peintre` a ,
 a ./Nom b , a ./Peintures/Peinture/@titre c ,
where $b = \text{"Van Gogh"}$

Slide 30

Traduction XQuery :

```

Q1(b):      FOR  $a IN document("http://...")/Peintre,
                $b IN $a/Nom,
                $c IN $a/Peinture/@titre
                WHERE $b = "Van Gogh"
                RETURN $c

```

Slide 31



Traduction de Requêtes

Binding : Mise en correspondance des chemins conceptuel de la requête avec des chemins conceptuels des règles de mapping :

Slide 32

- Concaténation des règles :
 - Un chemin de la requête peut être la concaténation de chemins de plusieurs requêtes
- Décomposition des requêtes :
 - Différentes sources répondent à différents fragments d'une requête
 - Clefs sémantiques :
 - * Absence d'identificateurs partagés entre les ressources
 - * Définition de clés au niveau médiateur



Décomposition

Requête abstraite :

Slide 33

```

select  d, f, g
from    Personne a, a.nom b, a.effectue.produit c,
          c.titre d, c.exposé_par e, e.nom f, e.city g
where   b = "Van Gogh"

```

Décomposition :

select <i>d</i>	select <i>f, g</i>
from <i>Personne a, a.nom b,</i>	from <i>Artefact c, c.exposé_par e,</i>
<i>a.effectue.produit c, c.titre d</i>	<i>e.nom f, e.city g</i>
where <i>b = "Van Gogh"</i>	



Clés sémantiques

Comment joindre les fragments obtenus par les deux sous-requêtes?

Requêtes clé : requête abstraite qui retourne pour un objet *o* une valeur de clé

Slide 34

```

select   t, a
from     Artefact o, o.titre t, o.produit.année a

```

Extension des requêtes sources :

```

select   d, t, y
from     Personne a, a.nom b, a.effectue.produit c,
          c.titre d, c.titre t, c.produit.année y
where    b = "Van Gogh"

```



Conclusion

Slide 35

- La médiation de requêtes est utilisée comme composant d'intégration d'entrepôts de données XML (médiation hybride).
- La médiation 'pure) à l'échelle du Web est limitée par des problèmes de performance et de sécurité.
- Mais, les nouvelles architectures (services Web, P2P, grid computing) pourraient changer cette situation ?
- Prochain cours : Conclusion et perspectives