
Intégration de données

Dan VODISLAV

Université de Cergy-Pontoise

Master Informatique M2

Plan

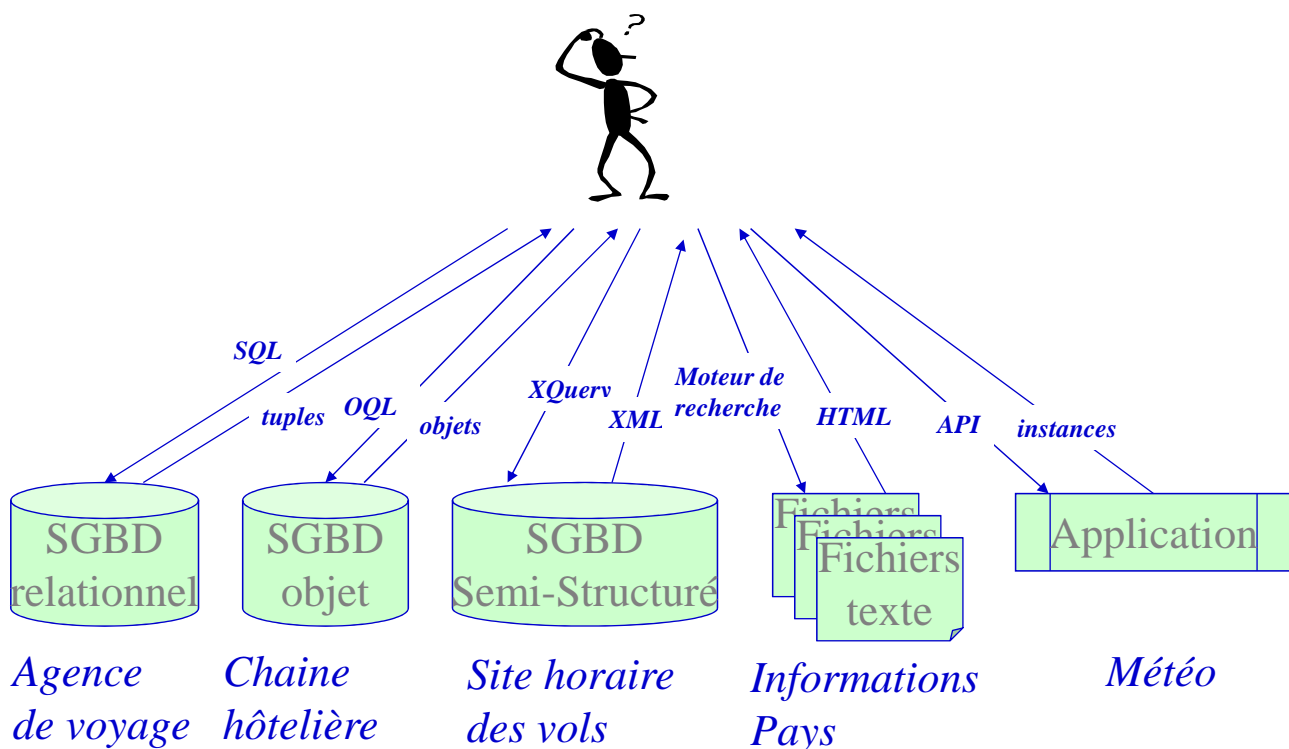
- Objectifs, principes, enjeux, applications
- Architectures d'intégration de données
 - Médiateurs et entrepôts
 - Traitement des requêtes
- Schémas d'intégration
 - Global-as-view
 - Local-as-view

Intégration de données

- Contexte
 - Sources d'information nombreuses et variées
 - SGBD relationnels/XML, pages HTML, LDAP, tableurs, fichiers, applications, ...
 - Interfaces d'accès variées en termes de:
 - Modèle de données: relationnel, XML, HTML, tableurs
 - Langages d'interrogation: SQL, XPath, XQuery, URL, ...
 - Protocoles de communication: JDBC, ODBC, SOAP, HTTP
- *Objectif général* : utiliser plusieurs sources comme si elles constituaient une seule base de données homogène → *l'intégration de données* doit fournir
 - *un accès* (requêtes, éventuellement mises-à-jour)
 - *uniforme* (comme si c'était une seule BD homogène)
 - *à des sources* (pas seulement des BD)
 - *multiples* (déjà deux est un problème)
 - *autonomes* (sans affecter leur comportement, indépendamment des autres sources ou du système d'intégration)
 - *hétérogènes* (différents modèles de données, schémas)
 - *structurées* (ou semi-structurées)

Page 3

Exemple



Page 4

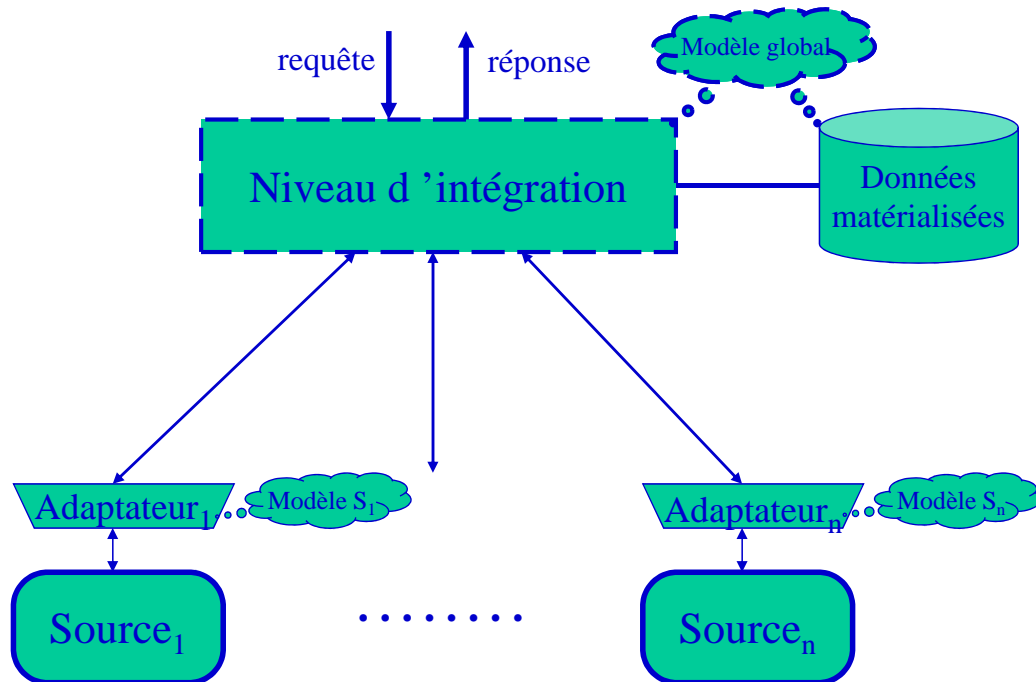
Applications

- Dans l'entreprise : « *business integration* »
 - Accès efficace, facile et sûr à des données internes et externes à l'entreprise
 - Données: BD locales, textes lois/contrats, applications CRM, eProcurement, ...
 - Une partie très importante des budgets IT sont dépensés en intégration
- Communautés web
 - Partage d'information produites par divers acteurs
 - Portail d'accès uniforme
 - Exemple: recherche biomédicale
- Grand public
 - Accès simple, rapide et efficace aux informations disponibles sur le web
 - Texte/HTML, images, vidéo, XML, fils RSS, cartes
 - Le web caché, services web
 - Commerce électronique: comparateurs de prix, intégration de magasins en ligne

Caractéristiques de l'intégration de données

- *Distribution*
 - Répartition géographique des sources sur le réseau
 - Échelle
- *Autonomie*
 - Les sources décident de ce qu'elles partagent, comment et quand
- *Hétérogénéité*
 - De format, de structure, de mode d'accès, de capacité de traitement

Architecture générale d'intégration



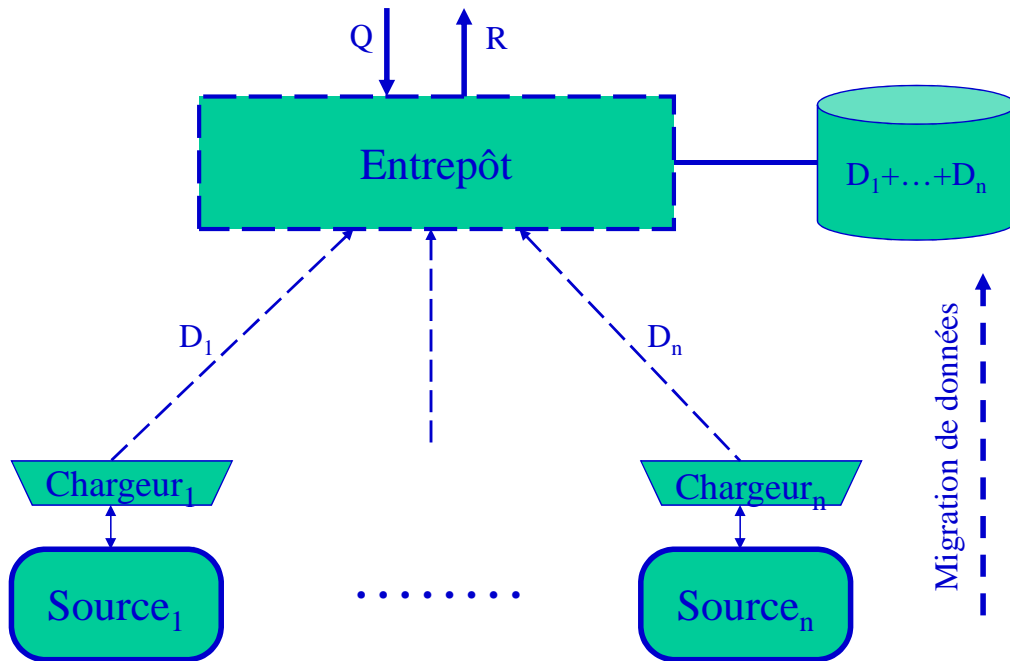
Page 7

Intégration matérialisée et virtuelle

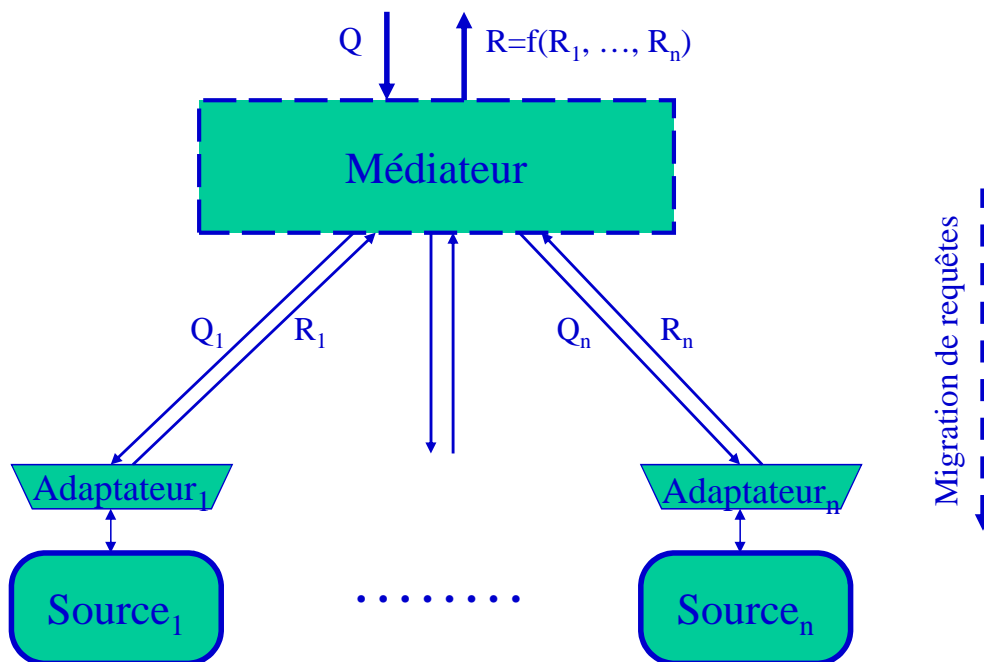
- Intégration matérialisée → *entrepôt de données*
 - Les données provenant des sources sont transformées et stockées sur un support spécifique (entrepôt de données).
 - L'interrogation s'effectue comme sur une BD classique
- Intégration virtuelle → *médiateur*
 - Les données restent dans les sources
 - Les requêtes sont exprimées sur le schéma global, puis décomposées en sous-requêtes sur les sources
 - Les résultats des sources sont combinés pour former le résultat final
- En pratique on peut avoir des architectures intermédiaires, entre ces deux extrêmes

Page 8

Architecture d'entrepôt



Architecture de médiation

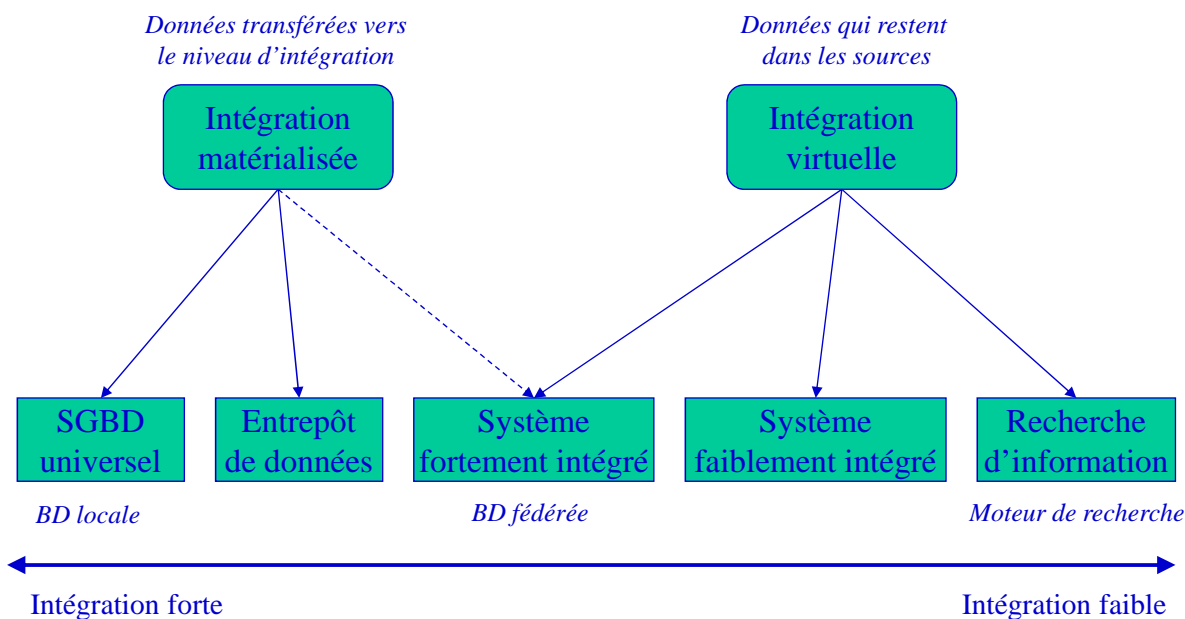


Entrepôt ou médiateur?

- **Médiateur** : accès direct aux sources
 - approche « paresseuse », pas de matérialisation
 - migration de requêtes vers les sources
 - *avantages* : données toujours fraîches, plus facile d'ajouter de nouvelles sources, plus grande échelle, distribution de l'effort
 - *inconvénients* : performances, traduction de requêtes, capacités différentes des sources
- **Entrepôt de données** : accès efficace à une copie des données
 - matérialisation des sources au niveau du modèle global
 - migration de données vers l'entrepôt
 - *avantages* : performances, personnalisation des données (nettoyage, filtrage), versions
 - *inconvénients* : données pas toujours fraîches, cohérence, gestion des mises-à-jour, gestion de gros volumes de données

Page 11

Degré d'intégration des données

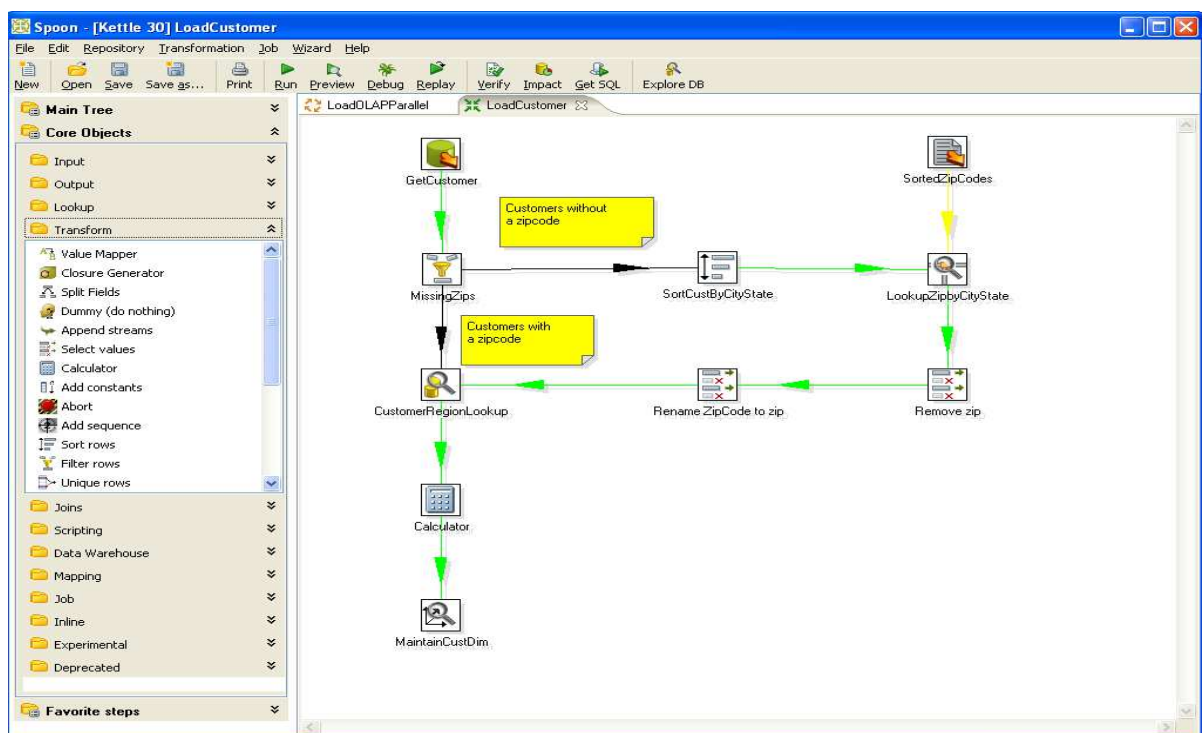


Page 12

Entrepôts de données

- L'approche la plus populaire d'intégration de données
 - Gros avantages: performances, contrôle plus facile à réaliser sur l'hétérogénéité des données
- Utilisation pour les systèmes décisionnels OLAP
- Transformation de données pour alimenter l'entrepôt
 - Chargeurs = *systèmes ETL* (« Extract, Transform, Load »)
 - Outils graphiques pour définir *des flots de traitements/transformations*
 - Une fois le flot de traitement défini → appliqué au contenu des sources

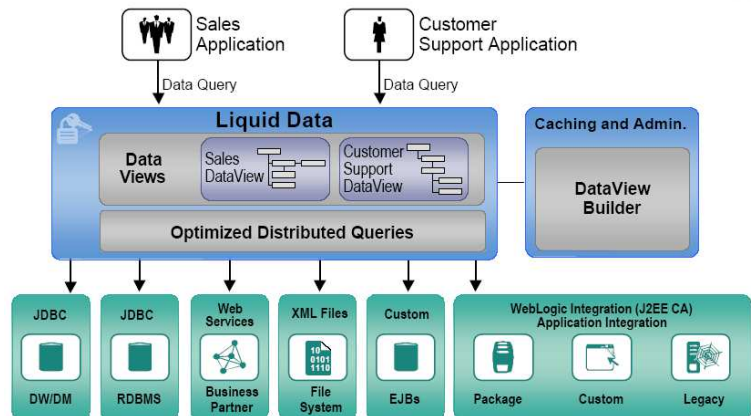
Exemple d'ETL : Kettle



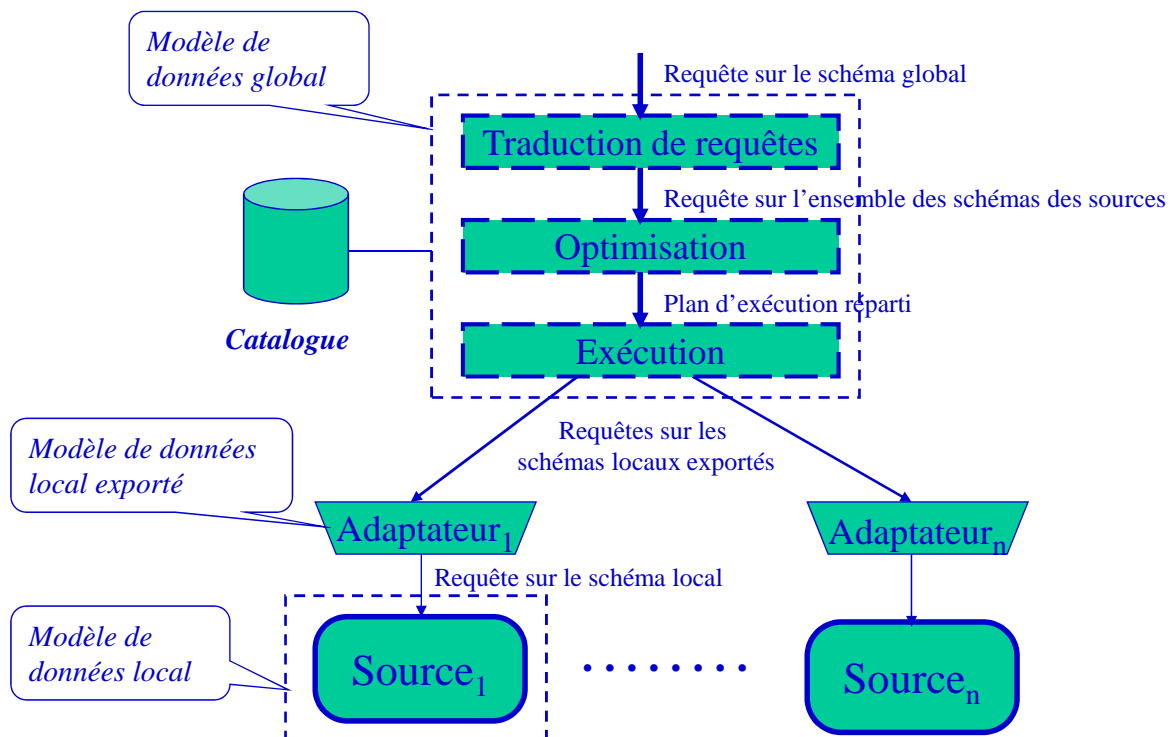
Médiateurs

- Bien que moins utilisés en pratique, ils ont plus de potentiel
 - Meilleur passage à l'échelle
 - Acceptent mieux les changements dynamiques (nouvelles sources)
- mieux adaptés à l'intégration de sources web

- En entreprise: EII
« Enterprise Information Integration »
 - Ex. BEA Liquid Data, IBM Websphere Information Integrator

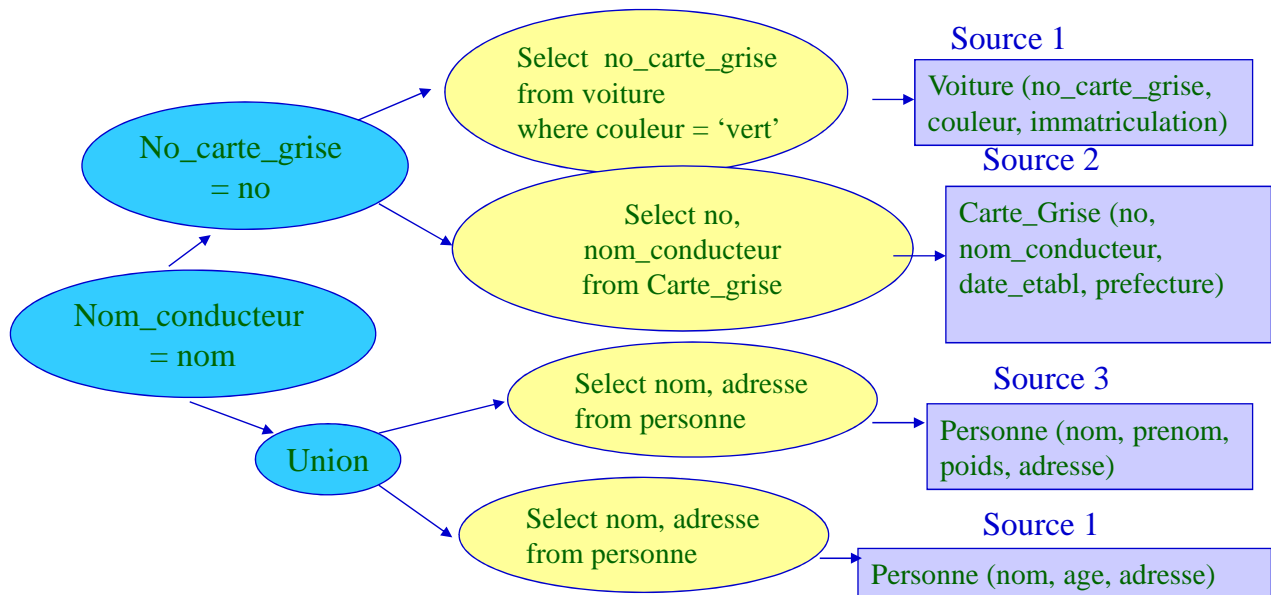


Architecture plus détaillée



Décomposition des requêtes

- Exemple : chercher l'adresse de tous les propriétaires de voiture verte



Page 17

Schémas d'intégration

- Problèmes
 - *Intégration de schéma*: comment définir un schéma global d'intégration à partir des schémas des sources?
 - *Fusion de données*: comment rendre compatibles, transformer les données en provenance des sources?
 - *Mappings/vue d'intégration*: comment décrire le lien entre le schéma global et les schémas des sources?

Page 18

Définition de la vue d'intégration

- Le lien entre schéma global et schémas locaux est défini à travers des vues
 - Mapping entre ces schémas
- Deux façons principales de définir ce lien
 - Le schéma global en fonction des schémas locaux → « *global as view* »
 - Approche *ascendante*: on part des sources pour produire le schéma global
 - Les schémas locaux en fonction du schéma global → « *local as view* »
 - Approche *descendante*: on fixe le schéma global et on décrit les sources par rapport à ce schéma fixé

Page 19

« Global-as-View »

- Le modèle global = vue sur les sources
 - élément global = f(éléments des sources)
 - $M = V(S_1, \dots, S_n)$
- Avantages
 - approche naturelle
 - la traduction de requêtes se fait facilement
- Inconvénients
 - nouvelle source → modification du modèle global
 - il faut considérer l'interaction de la nouvelle source avec les autres

Page 20

Systeme « Global-as-View »

• TSIMMIS (Stanford)

- Sources : informations sur les personnes d'une université
 - **Inf** : BDR avec des employés et des étudiants du département Informatique
Employé(Nom, Prénom, Titre, Chef)
Étudiant(Nom, Prénom, Année)
 - **Ann** : Annuaire pour l'université (nom, département, catégorie, e-mail, ...)
- Médiateur : les personnes du département Informatique
 - nom, catégorie, titre, chef, e-mail, année, ...
- langage de spécification de médiateur MSL
 - règles : $PM :- P_1, \dots, P_k$, avec PM, P_i « patterns »

TSIMMIS : modèle

Adaptateur Inf

```
<employe>
  <nom>Dupont</nom>
  <prenom>Michel</prenom>
  <titre>professeur</titre>
  <chef>Jean Martin</chef>
</employe>
<etudiant>
  <nom>Hugo</nom>
  <prenom>Victor</prenom>
  <annee>2</annee>
</etudiant> ...
```

Adaptateur Ann

```
<personne>
  <nom>Michel Dupont</nom>
  <dept>Informatique</dept>
  <categ>employé</categ>
  <email>md@univ.fr</email>
</personne>
<personne>
  <nom>Zoé Durand</nom>
  <dept>Informatique</dept>
  <categ>étudiant</categ>
  <annee>3</annee>
</personne> ...
```

Médiateur

```
<pers_inf>
  <nom>Michel Dupont</nom>
  <categorie>employé</categorie>
  <titre>professeur</titre>
  <chef>Jean Martin</chef>
  <email>md@univ.fr</email>
</pers_inf> ...
```

Spécification MSL du médiateur

```
<pers_inf>
  <nom>N</nom>
  <categorie>C</categorie>
  Reste1 Reste2
</pers_inf> :-
  <personne>
    <nom>N</nom> <dept>Informatique</dept>
    <categ>C</categ> Reste1
  </personne> @Ann AND
  <C>
    <nom>NF</nom> <prenom>P</prenom> Reste2
  </C> @Inf AND
  decomp(N, NF, P)
```

TSIMMIS : requêtes

- Exemple de requête

- trouver toutes les informations sur Michel Dupont

```
<pers_inf> <nom>Michel Dupont</nom></pers_inf>@Med
```

- substitution des éléments de la requête dans la définition du médiateur

```
<pers_inf> <nom>Michel Dupont</nom> <categorie>C</categorie> Reste1 Reste2 </pers_inf> :-
```

```
<personne>
```

```
  <nom>Michel Dupont</nom> <dept>Informatique</dept> <categ>C</categ> Reste1  
</personne>@Ann AND
```

```
<C>
```

```
  <nom>NF</nom><prenom>P</prenom> Reste2  
</C>@Inf AND
```

```
decomp("Michel Dupont", NF, P)
```

- chaque source répondra à la sous-requête qui la concerne

Page 23

« Local-as-View »

- Les sources = vues matérialisées du modèle global

- une source décrit les données du modèle global qu'elle peut fournir
- élément source = f(éléments modèle global)

$$S_i \subseteq V_i (M)$$

- Avantages

- les sources sont décrites indépendamment les unes des autres
 - très simple de rajouter une nouvelle source

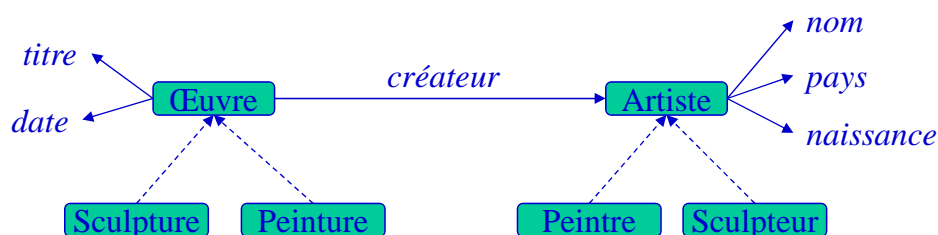
- Inconvénients

- traduction de requêtes plus complexe

Page 24

Système « Local-as-View »

- Information Manifold (AT&T)
 - modèle global : de type entité – association, exprimé par des relations
- Exemple de modèle global
 - Œuvre(titre, date, créateur), Artiste(nom, pays, naissance)
 - Sculpture, Peinture < Œuvre (sous-classes de Œuvre)
 - Peintre, Sculpteur < Artiste (sous-classes de Artiste)
 - Sculpture(titre, date, créateur), Peinture(titre, date, créateur),
 - Peintre(nom, pays, naissance), Sculpteur(nom, pays, naissance)



Page 25

Information Manifold : sources

- Sources : vues sur le modèle global
 - définition = requête conjonctive + inégalités
- Exemple de description de sources
 - S_1 : noms/dates naissance des peintres nés après 1800 et les titres/dates de leurs peintures
 $S_1(t, d, n, dn) \subseteq \text{Peintre}(n, p, dn), \text{Peinture}(t, d, n), dn \geq 1800$
 - S_2 : titres/dates des œuvres réalisées avant 1940 et le nom/pays de leurs auteurs
 $S_2(t, d, n, p) \subseteq \text{Œuvre}(t, d, n), \text{Artiste}(n, p, dn), d \leq 1940$
 - S_3 : noms et dates de naissance des sculpteurs français
 $S_3(n, dn) \subseteq \text{Sculpteur}(n, \text{'France'}, dn)$

Page 26

Information Manifold : requêtes

- Requête
 - titre/date des œuvres après 1900 + nom/date naissance de leurs créateurs
 $Q(t, d, n, dn) : \text{Œuvre}(t, d, n), \text{Artiste}(n, p, dn), d > 1900$
- Algorithme Bucket
 - identifier les sources pour chaque sous-requête (avec vérif. contraintes)
 - $\text{Œuvre}(t, d, n) : S_1(t, d, n, dn'), S_2(t, d, n, p')$
 - $\text{Artiste}(n, p?, dn) : S_1(t', d', n, dn), S_3(n, dn)$ (p inutile, S_2 ne fournit pas dn)
 - union de toutes les combinaisons valides des sources
 - on élimine les combinaisons qui produisent des résultats déjà obtenus par ailleurs
 $Q(t, d, n, dn) : S_1(t, d, n, dn), d > 1900$
 $Q(t, d, n, dn) : S_2(t, d, n, p'), S_3(n, dn), d > 1900$
Les combinaisons S_1 - S_3 et S_1 - S_2 sont plus restrictives que la solution S_1 seule !
- Remarque : dans GAV, les jointures entre sources sont déjà exprimées, dans LAV il faut les déduire

Page 27

Exercice « Global-as-View »

- Sources de données :
 - **S1**: *FilmTitre* (fid, titre)
FilmDetails (fid, année, genre, réalisateur)
Acteur (aid, nom, pays)
Joue (aid, fid)
 - **S2**: *FilmInfo* (titre, année, genre)
 - **S3**: *Realisateur* (nom, titre)
- Modèle global du médiateur GAV:
 - **M**: *Film* (titre, année, genre, réalisateur)
Acteur (nom, titre)
- Description GAV:
 - **M:Film**(t, a, g, r) :- **S1:FilmTitre**(fid, t) **S1:FilmDetails**(fid, a, g, r)
:- **S2:FilmInfo**(t, a, g) **S3:Realisateur**(r, t)
 - **M:Acteur**(n, t) :- **S1:FilmTitre**(fid, t) **S1:Joue**(aid, fid) **S1:Acteur**(aid, n, p)

Page 28

Requêtes GAV

- Exemple: les titres des films de James Cameron
 - $Q(t) = M:Film(t, a, g, "James\ Cameron")$
- Méthode: remplacement direct dans la description du médiateur
 - $M:Film(t, a, g, "James\ Cameron")$
 - :- $S1:FilmTitre(fid, t) S1:FilmDetails(fid, a, g, "James\ Cameron")$
 - :- $S2:FilmInfo(t, a, g) S3:Realisateur("James\ Cameron", t)$
- Remarque: l'interrogation de S2 est inutile!
- Résultat:
 - $Q(t) \rightarrow S1:FilmTitre(fid, t) FilmDetails(fid, a, g, "James\ Cameron") \cup S3:Realisateur("James\ Cameron", t)$

Exemple « Local-as-View »

- Modèle global du médiateur LAV:
 - $M: Film$ (titre, année, genre, réalisateur)
 - $Acteur$ (nom, titre)
- Sources de données :
 - $S1$: titre, années et réalisateurs de comédies
 - $S2$: acteurs avec le titre et année des films où ils ont joué
 - $S3$: réalisateurs, titres des films et acteurs qui y jouent
 - $S4$: titre et genre de films français d'avant 1970
- Description LAV:
 - $S1(t, a, r) \subseteq M:Film(t, a, "comédie", r)$
 - $S2(n, t, a) \subseteq M:Acteur(n, t) M:Film(t, a, g, r)$
 - $S3(r, t, n) \subseteq M:Film(t, a, g, r) M:Acteur(n, t)$
 - $S4(t, g) \subseteq M:Film(t, a, g, r), a < 1970$

Requêtes LAV

- Exemple: les acteurs dirigés par James Cameron
 - $Q(\mathbf{n}) = M:Acteur(\mathbf{n}, \underline{t}) M:Film(\underline{t}, a, g, \text{"James Cameron"})$
- Algorithme Bucket
 - identifier les sources pour chaque sous-requête
 - $Acteur(\mathbf{n}, t) : S_2(\mathbf{n}, t, a'), S_3(r', t, \mathbf{n})$
 - $Film(t, a?, g?, r) : S_1(t, a', r), S_3(r, t, \mathbf{n}')$ (a, g inutiles, S2/S4 n'ont pas r)
 - union de toutes les combinaisons valides des sources avec élimination des redondances
$$Q(\mathbf{n}) \rightarrow S_2(\mathbf{n}, \underline{t}, a') S_1(\underline{t}, a'', \text{"James Cameron"}) \cup S_3(\text{"James Cameron"}, t', \mathbf{n})$$

Les combinaisons S_2-S_3 et S_3-S_1 sont plus restrictives que la solution S_3 seule !
- Remarque : dans GAV, les jointures entre sources sont déjà exprimées, dans LAV il faut les déduire