École Doctorale 2008/2009

Systèmes de Bases de Données Avancés

6. Tendances actuelles

École nationale Supérieure d'Informatique

Plan

Les systèmes post-relationnels

Transactions Imbriquées

Quelques axes de recherche

6.1. Les systèmes post-relationnels

Modèles objets

- Types abstraits de données
  - Classes
  - Regroupement : données + traitement
- Liens
  - Héritage, Référence, Composition ...
- Attributs multi-valués
  - Collections
- Identifiant d'objets
  - OID système

Applications courantes

- SGBD Objet-Relationnel (extensibles)
  - Fonctions, Types, Opérateurs, Méthodes d'accès
  - SQL 99 (Extension du relationnel)
  - Exemple : PostgreSQL
    - http://www.postgresql.org/
- SGBD Orienté Objet
  - Langage POO + fonctionnalités SGBD
  - Modèle objet (ODL , OQL)
  - Exemple : EyeDB (http://www.eyedb.org/)

Exemples ODL

```
class Person {
  attribute string name;
  attribute Address addr;
  attribute Person * spouse inverse Person::spouse;
  attribute set<Car *> * cars inverse Car::owner;
  attribute Person *children[];

  instmethod void change_address (in string street, in string town,
                                  out string oldstreet, out string oldtown);
  classmethod int getPersonCount();
  index on name;
};

class Car {
  attribute string brand;
  attribute int num;
  attribute Person *owner inverse Person::cars;
};
```

Exemples OQL

select Person; // retourne les OIDs des instances de la classe Person

select Person.name = "xyz";
// retourne les personnes dont le nom est "xyz"

select x.name from Person x where x.age > 14 && x.age < 19;
// retourne le nom des personnes vérifiant la condition.

(select Person.name = "xyz").age := 20;
// mise à jour conditionnelle

```
for x in (select Person)
     if ( x.age < 50 )
          x.name := 'aaaa';
```

x := select Person.name = "xyz");
add Person(name : "enf1", age : 2) to first(x)->children;

Autres modèles

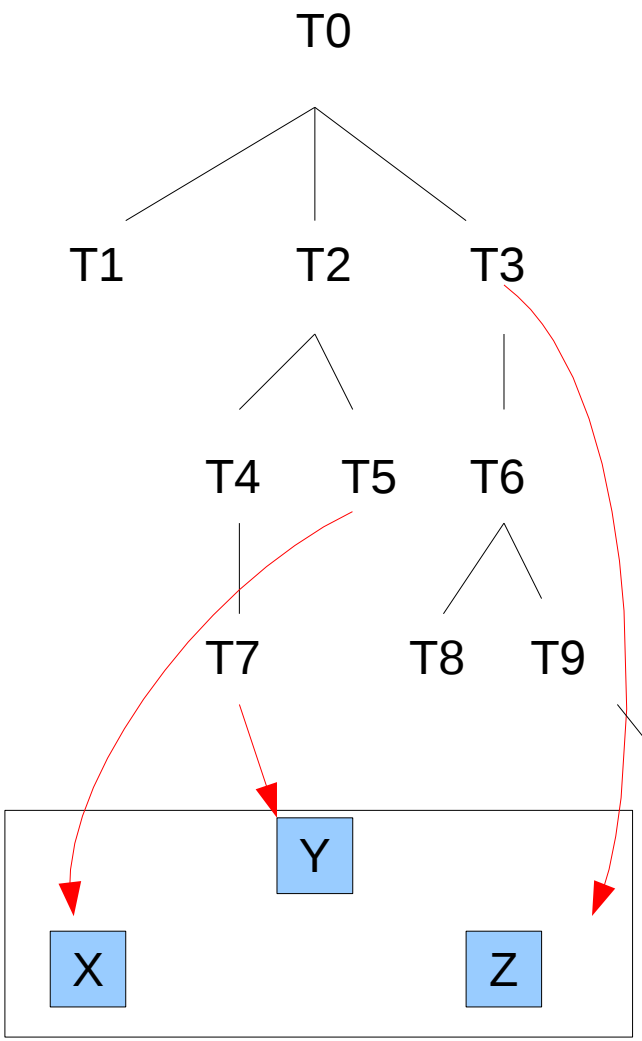
- NF2
  - « Non First Normal Form »
  - Relâchement de la 1ere forme normale
- Objets complexes
  - Constructeurs : Ensembles et Tuples
- Modèle fonctionnel
  - Les attributs sont vus comme des fonctions sur des objets
  - Gestion dynamique des types

6.2. Transactions Imbriquées

Transactions Imbriquées

- Pourquoi
  - Modèles objets
  - Pb des transactions longues
  - Plus de modularité
- Solution
  - Chaque transaction peut être composée de plusieurs sous-transactions (plus ou moins indépendantes)
  - Chaque sous-transaction peut aussi être composé de plusieurs sous-transactions.
  - Convient bien aux modèles objets
    - Méthode = Transaction Imbriquée

Transactions Imbriquées



Transactions racine (ACID)

Chaque transaction peut lancer plusieurs sous transaction (en séquentiel et/ou en parallèle)

une transaction ne peut **signaler sa terminaison** à sa transaction mère que si toutes ses sous transactions lui ont signalé leurs terminaisons

L'accès aux données se fait à partir de n'importe quel noeud de la hiérarchie

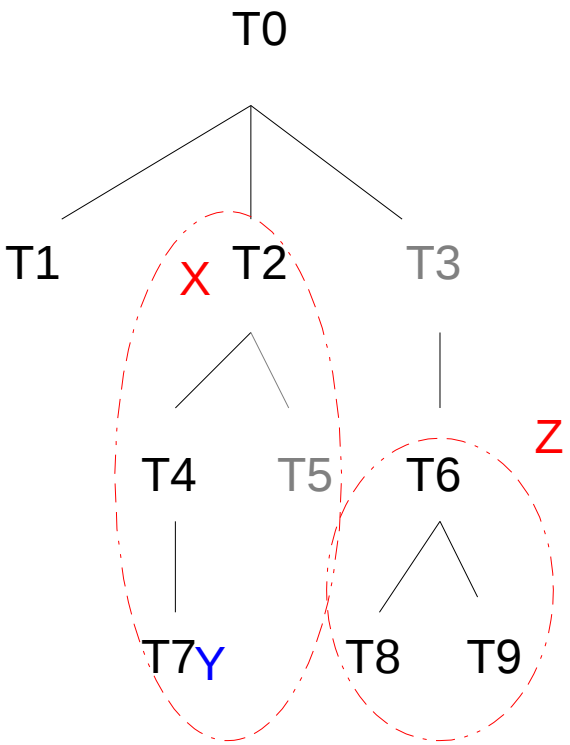
Concurrence dans le modèle imbriqué

- Terminaison
  - Si T est annulée, tous ses descendants le sont aussi
  - T ne peut être validée que si tous ses descendants ont terminé leur exécution (annulation ou pré-validation)
- Les verrous acquits par une transaction peuvent être transmis vers :
  - Sa transaction mère (héritage ascendant)
  - Ses transactions filles (héritage descendant)
- Une transaction peut donc :
  - détenir des verrous (acquisition normale des verrous)
  - hériter des verrous (qui lui sont réservés et à ses descendants)

Héritage de verrous

- Si une transaction hérite d'un verrou, celui-ci reste réservé pour cette transaction et ses descendantes
  - Une transaction en dehors de sa hiérarchie ne peut pas acquérir ce verrou, même si aucune transaction ne le possède encore
- L'héritage ascendant des verrous (détenus et hérités) peut être réalisé au moment où une transaction fille termine son exécution (par pré-validation)
- L'héritage descendant peut être contrôlé (durant l'exécution de la transaction mère) ou automatique (en fin d'exécution). Il ne concerne que les verrous détenus

Exemple



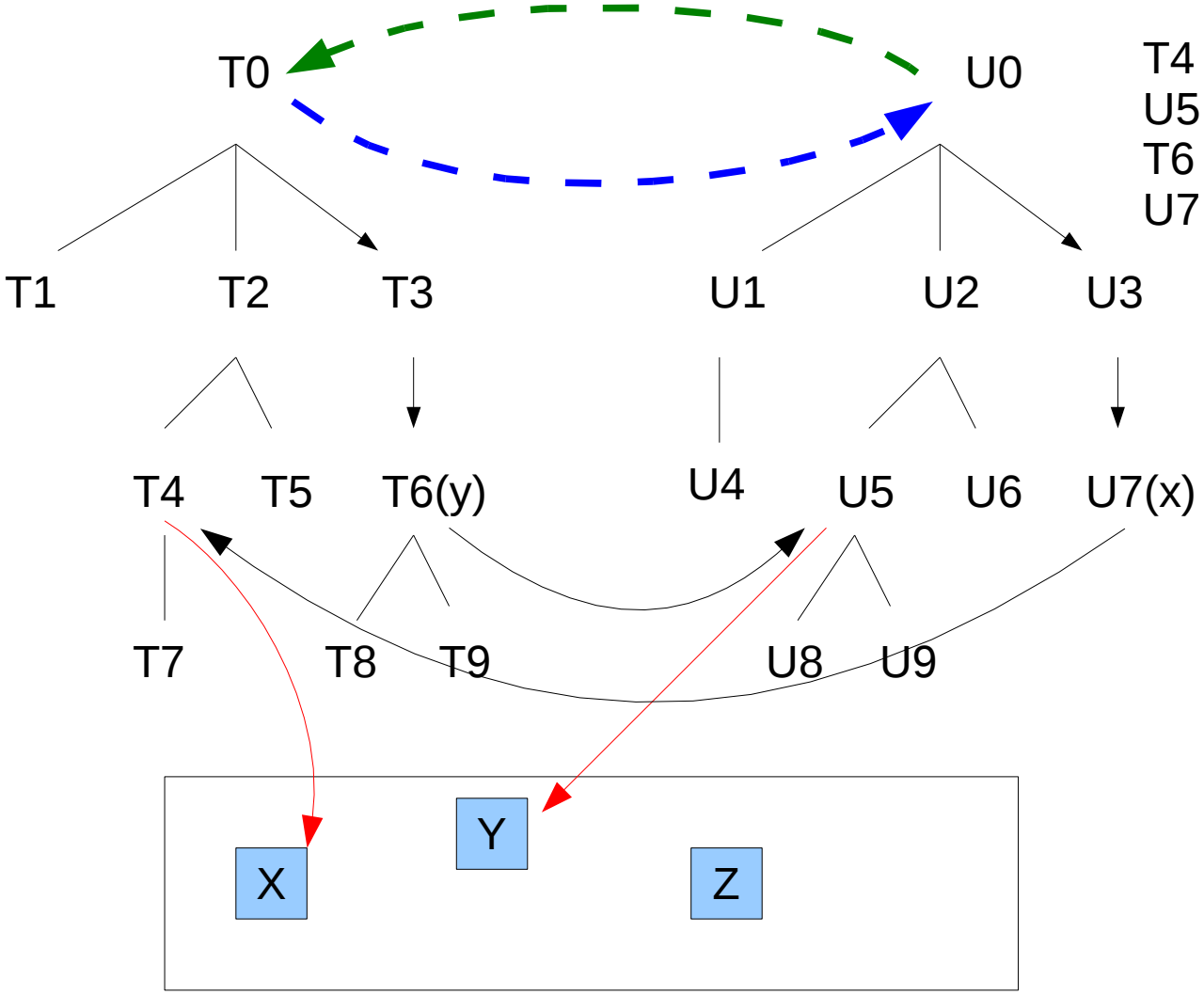
Exemple d'héritage ascendant:
T2 a hérité le verrou sur X de sa fille T5 (en phase de terminaison)
==> T2, T4 et T7 peuvent donc demander à acquérir le verrou sur X

Exemple d'héritage descendant:
T6 a hérité du verrou sur Z de sa mère T3 (en phase de terminaison)
==> T6, T8 et T9 peuvent donc demander à acquérir le verrou sur Z

Interblocages complexes

- Les différents types d'attente
  - Attente d'un verrou non libre (cas standard)
  - Attente de terminaison (wait-for-commit)
    - Une transaction mère doit attendre la terminaison de toutes ses transactions filles, avant de signaler sa propre terminaison.
  - Les liens de transitivité combinés entre les 2 premiers type d'attentes
- Le graphe des attentes contient donc des liens indirectes (ascendants, descendants et mixtes)
  - La détection de circuits est donc plus difficile

Exemple - interblocage



T4 détient un verrou sur X
U5 détient un verrou sur Y
T6 demande un verrou sur Y : attente
U7 demande un verrou sur X : attente
==> Interblocage

T0-T3-T6 (attente terminaison)
T6-U5 (demande de verrou)
T6-U2-U0 (hérit. ascendant)
T0-U0 (transitivité)

U0-U3-U7 (attente terminaison)
U7-T4 (demande de verrou)
U7-T2-T0 (hérit. ascendant)
U0-T0 (transitivité)

6.3. Quelques axes de recherche

Nouveaux types d'utilisation

- Internet
  - P2P et Grille de calcul
  - Hétérogénéité indépendance des sites
  - Intégration des sources et web sémantique
  - Requêtes à préférence et profile utilisateurs
- Systèmes temps-réel
  - Prise en comptes des dates d'échéances dans les transactions
- Manipulation multi-média

Les défis à relever (performances)

- Mélange de transactions longues et courtes
- Validation atomique (extensions du 2PC)
- Haute disponibilité et recouvrement MMDB
- Gestion des copies et de l'inconsistance
- Efficacité des modèles riches

Exemple : Act21

- Un SGBD parallèle en RAM
- Modèle à base d'acteurs
  - Objets actifs et concurrents
- Stockage à base de SDDS → CTH*
  - Partitionnement dynamique et équilibré
- Transactions Imbriquées
  - Héritage ascendant et descendant des verrous
- Projet de recherche à l'INI (2002-2005)