Données ½ Structurées & XML (Partie II)

Master INFORMATIQUE
Nacer.Boudjlida@loria.fr
UdLorraine - LORIA

Données ½ Structurées & XML

- Les Problèmes:
  - Transformation/fusion/intégration de données fortement structurées
  - Idem pour des données non ou semi-structurées
  - XML, un cas particulier de données non structurées?

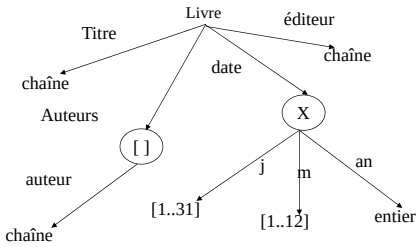
Données fortement structurées

Note : Arbre de données (Type et instance)

- Feuilles : types de base
- Nœuds internes : constructeurs de types
- Arcs labellés par les « noms des attributs » (fonctions de projection)
- Notations :
  - <X1 : T1, …, Xn : Tn> : produit cartésien (record)
  - [E : T] : liste d'éléments E de type T
  - * (E : T): ensemble d'éléments E de type T
  - [i..j] : intervalle

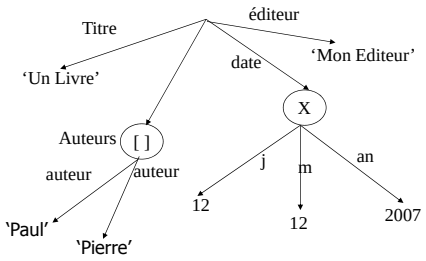
Données fortement structurées

- Livre :< Titre: chaîne, Auteurs: [auteur: chaîne], date : < j: [1..31], m: [1..12], an: entier>, éditeur : chaîne>
- Arbre du Type:



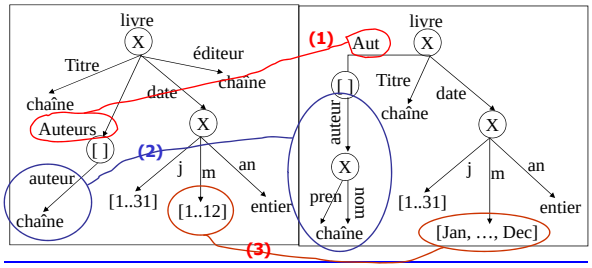
Données fortement structurées

- Arbre de l'instance du type:



Données fortement structurées (suite)

- Intégration et « Conflits » [cf. Sheth92]
  - Nommage (1) , Structures (2)
  - Domaines (3) , etc.



Données fortement structurées (suite)

- Processus d'intégration :
  1. Parcours/Comparaison des structures
  2. Résolution des conflits
  3. Intégration
     - Restructuration
     - Conversion des valeurs

Données fortement structurées (suite)

1. Comparaison

- Nommage : 'Aut' → 'Auteurs'
- Structure : <auteur : chaîne> → <auteur : <nom: chaîne, prénom : chaîne>>
- Domaine : <m : [1..12]> → <m : [Jan..Dec]>
- ???? : <éditeur : chaîne> → ∅

Données fortement structurées (suite)

2. Résolution des conflits

- Aut → Auteurs
- <auteur : chaîne> → <auteur : <nom: chaîne, prénom : chaîne>>
- <m : [1..12]> → <m : [Jan..Dec]>
- <éditeur : chaîne> → ∅

Données fortement structurées (suite)

3. Intégration

3.a. « Ré-écriture » des structures

- <Auteurs: [auteur: chaîne], Titre: chaîne, … >

  → <Titre: chaîne, Aut: [auteur: <nom, pren: chaîne>]..>

- X(A1: [A2: T2], A3: T3, …)

  → X(A3: T3, B1: [A2: X(B2: T4, B3: T5)], …)

- Sorte d'algèbre sur les constructeurs de types

Données fortement structurées (suite)

3.b. « Ré-écriture » des valeurs

- <12, 12, 1992 > → <12, Déc, 1992 >
- auteur : chaîne → auteur : <nom , pren : chaîne> ?
- [A, B, C, D] → [0..20] ?

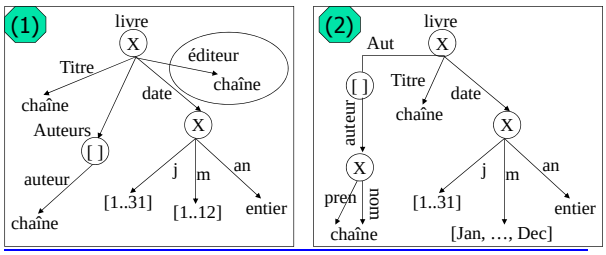
- Nombre fini de constructeurs de types
- Nombre fini de règles de ré-écriture de structures
- Souvent, solutions ad hoc

Note : Intégration de données vs de schémas

- Intégration de données :
  - Structure: Type 1 → Type 2
  - Valeur: Instances de Type 1 → Instances de Type 2
- Intégration/Fusion de schémas (≈ Type complexe)
  - n * Schémas → 1 Schéma
  - Instances conformes a chacun des n schémas → Instances conformes au schéma résultat

Données fortement structurées (suite)

- Intégration données vs Intégration schémas
  - (1) dans (2) | (2) dans (1)
  - (1) '∪' (2) =



Données fortement structurées (fin)

- Méta-données supposées disponibles
- Modèle commun de représentation: constructeurs de types et graphe de données
- Domaines des valeurs (généralement) simples
- *Cas* :
  - Pas de méta-données
  - Valeurs complexes (texte, sons, images, etc.)
  - Valeurs « fragmentées » et distribuées (ex: Web)
- Mêmes types de problèmes et de mécanismes?

Résumé

- Transformation de données structurées
  - Arbre/Graphe de données
  - Processus
    - Comparaison
    - Identification des ressemblances/dissemblances
    - Transformation
  - Opération sur
    - Structure (Type)
    - Instance (Valeur)
  - (Souvent) Solutions *ad hoc*

Données Semi-Structurées

- La suite:
  - Transformation de données fortement structurées
  - Transformation de données non ou semi-structurées
  - XML, un cas particulier de données non structurées

Données Semi-Structurées

- Pas nécessairement la même structure pour chaque instance
- Délibérément
  - « Oublier « le type de l'instance
  - Sérialiser les valeurs en annotant chaque élément par sa description : Données « auto-décrites »

Données Semi-Structurées (suite)

- *Exemple* : BDOO

{{*personne* : &o1{*nom* : LeBeauf, *Age* : 50,
enft1 : &o2, *enft2* : &03},

{personne : &o2{*prénom* : Gentil, *Age* : 23,
rel :{*mère* : &o1, *sœur* : &o3}}

{*personne* : &o3{*prénom* :Manon,
pays-naiss : Là-Bas, *mère* : &01}
}

Notion de graphe de données (suite)

- Graphe de données « objets »

Graphe de données (suite)

- &O_i : OID, Identifiant d 'objet (logique, physique, URL, etc.)
- *Dans la suite* : Nœuds avec/sans Oid :

  {a: &o1 {b: &o2 49}} isomorphe à

  {a: {b: 49}} et à {a: &o1{b: 49}}

Graphe de données (suite)

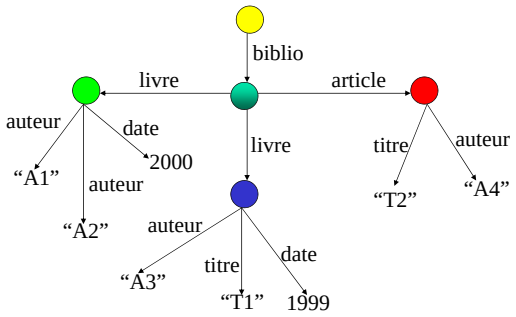
- {biblio:

  {livre: {auteur: `A1', auteur: `A2', date: 2000},

  livre: {auteur: `A3', titre: `T1', date: 1999},

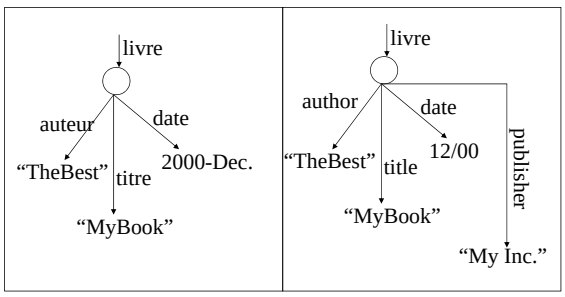
  article: {titre: `T3', auteur: `A4'} } }

Récursion structurelle : Cas des arbres



Récursion structurelle : Cas des arbres

- S 'agit-il des mêmes objets ?



Données semi-structurées: Récursion structurelle

- Egalité/Comparaison/Fusion de graphes de données
- Transformations complexes sur données semi-structurées par :
  - Parcours du graphe de données
  - Construction éventuelle d'un nouveau graphe

1. Cas des arbres (Récursion structurelle)

2. Cas des données cycliques

Récursion structurelle : Cas des arbres

- Récursion jusqu 'a atteindre une feuille: résultat = v ou { }
- { } dénote un graphe a un seul nœud (feuille)
- Quand une racine a plusieurs fils : décomposition en unions d 'arbres
  - t1 U t2 = {Val(t1), Val(t2)}

Récursion structurelle : Cas des arbres

- *Trouver tous les entiers*

1.
2.
3.
4.

Récursion structurelle : Cas des arbres

- *Exécuter l'algorithme précédent*
- *Corrigé*

Récursion structurelle : Cas des arbres

- Exercice: *Ecrire un algorithme qui permet de ré-écrire les entiers en chaînes. On dispose de deux fonctions:*
  - *Est_entier(x) = Vrai si x est un entier*
  - *Int2String(x) : Transforme l'entier x en chaîne de caractères*
    (Int2Sring(1) = 'UN'; Int2String(100) = 'CENT')

Récursion structurelle : Cas des arbres

- *Ré-écrire les entiers en chaînes*
- Corrigé

Récursion structurelle : Cas des arbres

- Modèle strict de définition :
  - Pour les valeurs atomiques (Déf. 1)
  - Pour { } (Déf. 2)
  - Pour des nœuds de type {l: t} (Déf. 3)
  - Pour t1 U t2 (Déf. 4)
- Pour { } : Résultat = { }, toujours
- Pour t1 U t2 : Résultat = f(t1) U f(t2)

Résumé

Transformation de données non structurées

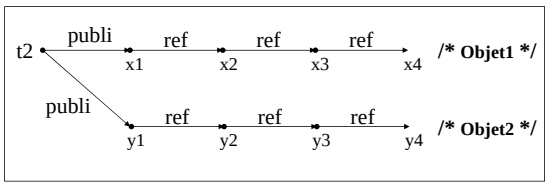
- Objets auto-décrits
- Pas de structure (typage) forte
- Arbres/Graphes de données
- Données non cycliques (Récursion structurelle)
- A suivre: Données cycliques (Structures Récursives)

2. Cas des données cycliques

- Exemple de données cycliques

Comparaison de données cycliques (suite)

- Considérer le dépliage « infini » des données
- Si x1 = y1 , …, xn = yn alors Objet1 = Objet2



Comparaison de données cycliques (suite)

- Deux objets sont égaux si on peut récursivement montrer que leurs arbres de dépliage sont égaux
- Dépliages infinis *mais* chaque dépliage a un nombre fini de sous-arbres *(arbres rationnels/réguliers)*
- Bi-simulation : Egalité sans dépliage

Comparaison de données cycliques (suite)

- x, x' : Nœuds de t1; y, y' : Nœuds de t2
- Relation de bi-simulation entre les nœuds des graphes (x ~ y) satisfait :

1.

2.

Comparaison de données cycliques (suite)

3.

4.

Comparaison de données cycliques (suite)

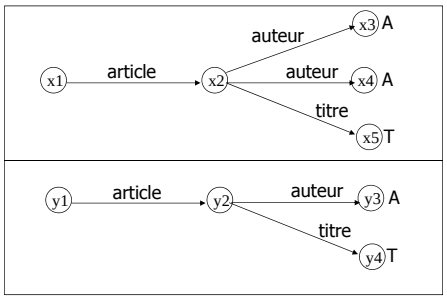
- Calcul de la relation, si elle existe

Comparaison de données cycliques (fin)

- *Calcul de la relation :*
  - Si n1, n2 sont le nombre de nœuds de t1, t2 : n1*n2 itérations
  - Chaque itération : inspection de tous les couples de nœuds des graphes : $O(n1^2 * n2^2)$

Comparaison de données cycliques (suite)

- *Exemple: Comparer les structures (corrigé)*



2. Données semi-structurées : Conclusion

- Structures
  - Irrégulières
  - Implicites (extraction ?)
  - Incomplètes
- Absence ou faible typage
- Exemples : OEM, LOREL
- XML : même veine ?

Données ½ structurées et XML

- La suite:
  - Transformation de données fortement structurées
  - Transformation de données non ou semi-structurées
  - XML, un cas particulier de données non structurées

3. Introduction à XML et « sa galaxie »

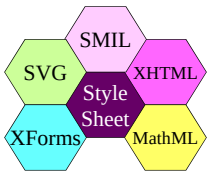
- XML: eXtendible Mark-up Language
  - World Wide Web Consortium (www.w3c.org)
  - Sous-ensemble de SGML (Standard Generalized Mark-up Language) : Gestion Electronique de Documents
  - Description du contenu des documents
  - Objectif Principal : Echange de données
  - Marquage (balisage) « libre » et extensible
  - Hypothèse : Compréhension commune des balises (e.g. XMI pour l'échange de diagrammes UML)

Introduction à XML

- Données *a priori* non interprétées (**P**arsed **C**haracter **DATA**)
- Langage facilement analysable
- Mise en forme du contenu (XML → HTML)
  - **CSS** (Cascading Style Sheet) : cf. HTML
  - **XSL** (eXtensible StyleSheet Language) : non reconnu comme standard
  - **XSLT**ransformation ≈ XSL Microsoft (Internet Explorer)
- Modules, Document profiles, DTDs (→ XML Schémas), …
- Génération de browsers Web fondée sur XML?

Introduction à XML

- XHTML
- But : être utilisable avec des balises provenant d'autres « vocabulaires » XML, comme
  - SMIL : Synchronized Multi-Média Integration Language
  - SVG : Scalable Vector Graphics
  - XForms : Formulaires Web
    - OFX : Open Financial eXchange

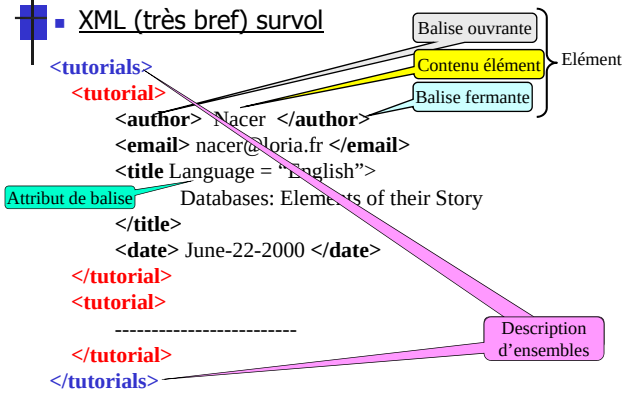


Introduction à XML

- XML :: HTML
  - Description du contenu
  - Pas d'ensemble prédéfini de balises
  - Support de structures imbriquées ≈ objets complexes (Xlink et Xpointer) :
    - Liens intra et inter documents
    - Navigation dans des documents XML
  - Données éventuellement auto-décrites :
    - **D**ocument **T**ype **D**efinitions
    - **D**ocument **C**ontent **D**escription (≈ Typage de document)
    - **R**esource **D**escription **F**ramework (pour Méta-données)
    - XML-Schema

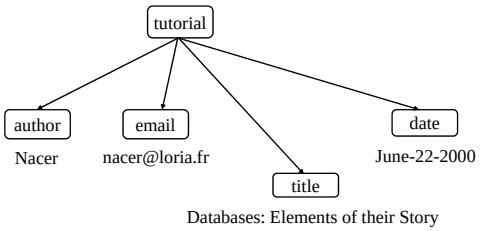
Introduction à XML

- XML (très bref) survol



Introduction à XML : Survol de XML (suite)

- Représentation :



Introduction à XML : Survol de XML (fin)

- XML :: Données semi-structurées:
  - ≈ Graphe de données auto-décrites
  - ≠ Labels dans les nœuds (:: sur les arcs)
  - ≠ Ordre sur les éléments
  - ≠ Processing Instructions, Commentaires, etc.
- XML : pour l'échange (Constante évolution)

XML : des adresses utiles

- Consortium W3 (tout sur XML et même plus):
  http://www.w3.org/
- Outils divers pour XML
  http://www.garshol.priv.no/dowload/xmltools

Web et Bases de Données

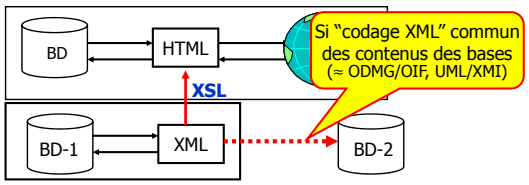
Web et Bases de données

1. Web en tant que bases de données
2. Introduction à Xquery
3. Conclusion

Web en tant que BdD

≠ Accès à des bases via le Web

- Langages de scripts (perl, php, etc.)
- Convertisseurs de formats (médiateurs):
  - Peer-To-Peer vs Représentation Commune
  - Ce qui est ou n'est pas (commercialement) disponible :



Web en tant que BdD

- Le Web comme base de données : pourquoi ?
  - Sources d'informations
  - Introduction d'un peu « d'organisation »
- Interrogation base de données : schéma nécessaire (!?)
- A priori, aucune structure (Web ≈ gigantesque graphe) : quel(s) modèle(s) de données ?

Langages d'interrogation pour le Web (1/3)

- WebSQL: Web modélisé comme une base relationnelle avec 2 relations (virtuelles) i.e. graphe d'objets atomiques
  - Document(url, title, …)
  - Anchor (Base, label, href)
- Exemple: « Documents sur Nancy dans les serveurs en France »

```
SELECT d.url, d.title
FROM Document d SUCH THAT d MENTIONS "Nancy"
WHERE d.url CONTAINS ".fr"
```


Langages d'interrogation pour le Web (2/3)

- WebOQL: Web modélisé comme un graphe d'objets structurés
- Interrogation :
  - Le Web
  - Une Page
  - Un ensemble de pages liées entre elles
- Restructuration :
  - HTML-HTML
  - HTML-Base de données
  - Base de données-HTML

Langages d'interrogation pour XML (3/3)

- XQuery : Influences
  - Quilt : langage de requêtes pour XML
  - SQL : Modèle « Select from where »
  - Xpath 1.0 (*) et XQL : Expressions de chemins
  - XML-QL : « binding » de variables
  - OQL : aspect fonctionnel du langage avec expressions imbriquées

(*) Notation pour naviguer dans un document XML

Langages d'interrogation pour XML (3/3)

- XQuery : les exigences (requirements)
  - Non procédural
  - Syntaxe en XML et Syntaxe « human readable »
  - Cloture du langage
  - E/S de XQuery : instances de XQuery Data Model
  - Combinaison d'opérateurs y compris des requêtes comme opérandes
  - Possibilité de combiner des données de sources différentes
  - Agrégation et tri

Langages d'interrogation pour XML (3/3)

- XQuery : les exigences (requirements)
  - Accès aux méta-données (Schéma, DTD)
  - Extensibilité (types, fonctions)
  - Applicabilité à tout type de sources XML i.e.
    - Documents stockés en mode natif ou
    - Documents vus comme des documents XML (grâce à des médiateurs ou des convertisseurs de formats)
- La suite :
  1. XML-QL : une des influences de XQuery
  2. XQuery : des exemples

Langages d'interrogation pour XML: XML-QL

- Relationnellement complet, pour des données de type relationnel
- Principaux composants du langage:
  - **where** ≈ clauses from + where de SQL
- Exemple:

3.2- Langages d'interrogation pour XML: XML-QL

- Intérêt des « patterns » :
  - Spécification du « modèle » de résultat
- Traitement des requêtes :
  - « Comparer » le pattern aux données (cf. comparaison de graphes/d'arbres de données semi-structurées)
  - Lier les variables
  - Retourner les variables de *construct*

Langages d'interrogation pour XML: XML-QL

- Spécification de la structure du résultat :

```
<query>
  where
    <freedb>
       <author> $A </author>
       <dbms>   $D         </dbms>
       <generation> OO </generation>
    </freedb>  in "www.acm.org/..."
  construct
     <result> <author> $A </author>
              <dbms> $D </dbms>
     </result>
</query>
```

```
<result>
   <author>
     <name> --- </name>
   </author>
   <dbms> --- </dbms>
</result>
      .........
<result>
      -------
</result>
```

Langages d'interrogation pour XML: XML-QL

- Prise en compte des différences de structures
  - Pas le même ensemble de balises
- Exemple: <release> non présent partout
  - SGBD et leur release, quand elle existe

```
where <freedb> $F </freedb> in "www.acm.org/...",
      <dbms> $D </> in $F
construct
      <result> <dbms> $D </dbms>
            where <release> $R </> in $F
            construct <release> $R </release>
      </result>
```


Langages d'interrogation pour XML: XQuery

- Principaux types d'expression XQuery
  - Expressions de chemin
  - Expressions For Let Where Return (*FLoWeR*)
  - Expressions avec opérateurs et fonctions
  - Expressions conditionnelles
  - Expressions avec quantificateurs
  - Expressions testant ou modifiant les types de données

3.2- Langages d'interrogation pour XML: XQuery

- Expressions de Chemins: Prédicats filtrant un ensemble de nœuds
  - document("fichier.xml") : nœud racine
  - **.** équivaut à *self::node()*
  - **..** équivaut à *parent::node()*
  - *nom* équivaut à *child::nom (/)*
  - *@nom* équivaut à *attribut::nom*
  - // équivaut à */descendant-or-self::node()/*

3.2- Langages d'interrogation pour XML: XQuery

- ***document****("zoo.xml")//chapter[2]//figure[caption = "Tree Frogs"]*
  - Nœud racine : zoo.xml
  - 2ème descendant du nœud racine : <chapter[2]>
  - Recherche des éléments <figure> n'importe où dans l'élément <chapter>
  - Sélectionner ceux ayant un attribut <caption> avec la valeur "Tree Frogs".
- ***document****("zoo.xml")//chapter[2* ***TO*** *5]//figure*

3.2- Langages d'interrogation pour XML: XQuery

- [**FOR** | **LET**] … **WHERE** … **RETURN**

```
FOR $b IN
  document("bib.xml")//book
WHERE $b/publisher = "Morgan
  Kaufmann"
AND $b/year = "1998"
RETURN $b/title

LET $b := document("bib.xml")//book
RETURN $b/title
      [unordered()|distinct()]
```


3.2- XML (fin) : des adresses utiles

- Consortium W3 (tout sur XML et même plus):
  http://www.w3.org/
- Outils divers pour XML
  http://www.garshol.priv.no/dowload/xmltools
- Implantations de XQuery
  http://www.softwareag.com/developer/quip
  http://www.xhive.com/

Conclusion : Bases de données vs Web

- Bases de données
  - Distinction entre les niveaux logiques et physiques
  - Données structurées et fortement typées (schémas)
  - Langages de description et de manipulation
  - Concurrence, etc.
- Sources Web:
  - Données non ou semi-structurées ("typage faible")
  - (Souvent) pas de schéma (méta-données)
  - Pas (encore) de langage d'interrogation

Conclusion : Bases de données vs Web (suite)

- Modèles de données
  - Objets, ODMG
- Stockage
  - Fichiers texte
  - Base relationnelle/objet
  - Graphes/Arbres
- Indexation
  - cf. objets complexes
  - Maintenance des index?
- Schéma de données
  - Bases sans schéma ?
  - Extraire le schéma des données?
  - Reverse engineering (HTML to XML schema) ?

Conclusion : Bases de données vs Web (suite)

- Langage de requêtes
  - SQL-Like, OQL
  - Comparaison des Objets
  - Traitement des requêtes
  - Portée des requêtes ?
    - Xquery
    - Egalité profonde/de surface
    - Cf. requêtes distribuées
    - Avec/sans méta données ?
    - Cycles données/hyperliens?
    - Une fédération de sources?
    - World-wide ?
    - Complexité et efficacité ?
    - Image, son, ...
- Hétérogénéité

Conclusion : Bases de données vs Web (suite)

- Optimisation de requêtes ?
- Mises à jour des sources ?
- Transactions ? Reprise ?
- Data mining vs Web mining
- Intégration Schémas/Vues
- Intégration de données hétérogènes

Du Relationnel au Web: Conclusions

- Noyau de la technologie BdD comme base (conceptuelle) : Besoin d'adaptation et d'extensions
- Ne pas "ré-inventer la roue"
- D. Suciu (Keynote address, 7th Intern'l Conf. On Database Systems for Advanced Applications, "Web Data and the Resurrection of Database Theory):

  "**new artefacts are not concepts but standards**"
- Nouveaux défis : hétérogénéité (données et structures), efficacité (stockage, requêtes), etc.

Données ½ Structurées & XML : La suite

- Données structurées versus données semi-structurées
- XML et une partie de sa « galaxie »
  - Standards autour de XML: XSLT , XLINK, XPOINTER, ...
  - Métadonnées, Grammaire et Schema (XML-Namespace, DTD, XML-Schema)
  - Localisation et évaluation de requêtes (XPath et XQuery)
  - XML et les Bases de Données, orientation documents et données
- Sécurité
  - Dans les Bases de Données
  - Modèles de contrôle d'accès pour les Bases de Données XML.

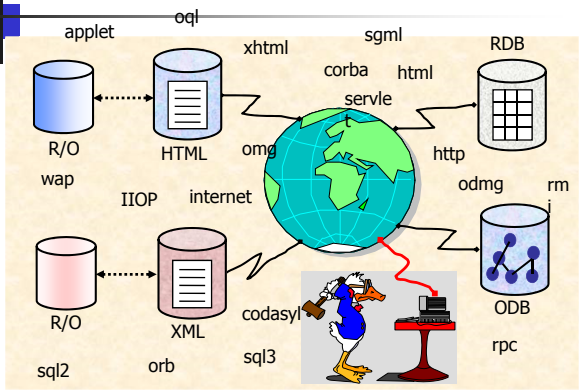
Du Relationnel au Web : C'est la fin

That's all Folks !

Merci de votre attention.

Questions ?

Conclusion : Bases de données vs Web (fin)



Eléments de bibliographie

- K. Sivashanmugam, K. Verma A. Sheth and J. Miller, Adding Semantic to Web Services Standards, *1rst Intern'l Conference on Web Services (ICWS'03),* Las Vegas, Nevada, June 2003, pp. 395-401
- A. Kiryakov, B. Popov, I. Terziev, D. Manov, D. Ognyanoff, Semantic Annotation, Indexing, and Retrieval. *Elsevier's Journal of Web Semantics*, Vol. 2, Issue (1), 2005.
- N. Boudjlida and D. Cheng, Complement Concept and Capability Discovery, *in proceedings of the CaiSE (Conference on Advanced Information Systems Engineering) EMOI'O4 (Enterprise Modelling and Ontologies for Interoperability) Open Workshop*, Vol.3, p.337-342, J. Grundspenkis and M. Kirikova Editors, Riga, Latvia, 7-11 June 2004.
- Silvana Castano, Maria Grazia Fugini, Giancarlo Martella, Pierangela Samarati, *Database Security*, Addison-Wesley & ACM Press 1995.

Eléments de bibliographie

- N. Boudjlida and H. Panetto, editors. *Proceedings of the 2nd InternationalWorkshop on Enterprises and Neworked Enterprises Interoperability (ENEI)* in J. Eder and S. Dustdar eds, Business Process Management Workshops, BPM 2006 International Workshops, Vienna, Austria, September 2006, LNCS# 4103, Springer-Verlag, June 2006. ISBN 10: 3-540-38444-8.
- Serge Abiteboul, Peter Buneman, Dan Suciu, '*Data on the Web: From Relations to Semistructured Data and XML*', Morgan Kaufmann Publishers, October 1999.
- J. Krogstie, C. Rolland, and N. Boudjlida, editors. *Journal of Enterprise Information Systems, Special Issue on Interoperability of Enterprise Systems and Applications*, 2006. Vol. 3, Issue 1.
- N. Boudjlida and H. Panetto. *Enterprise semantic modelling for interoperability*. In Proceedings of the 12th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2007, 25-28 September, Patras, Greece, 2007.