Le Web de Données

Dan VODISLAV

CY Cergy Paris Université

Master Informatique M1

Cours IED

Plan

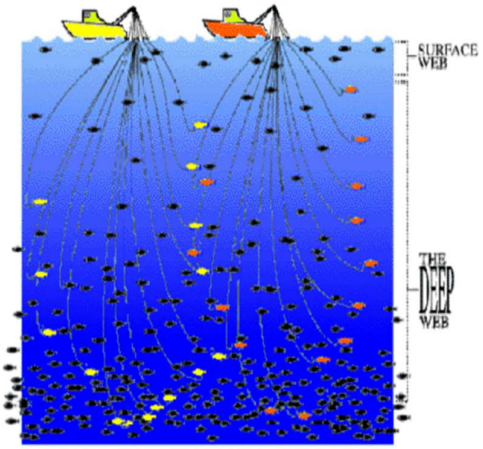
- Evolution du Web
- RDF sur le Web
  - Micro-formats
  - Micro-données
  - RDFa
- Vocabulaires communs
  - Dublin Core, FOAF, SKOS
- Linked Open Data
  - Architecture LOD
  - Nuage LOD
  - Points d'accès SPARQL

L'évolution du Web

- Web 1.0
  - Contenu non structuré (texte/HTML)
  - Consommateurs passifs
- Web 2.0
  - Contenus plus structurés (XML, JSON)
  - Consommateurs actifs
  - Quelques sites gérant de gros volumes de contenu spécialisé

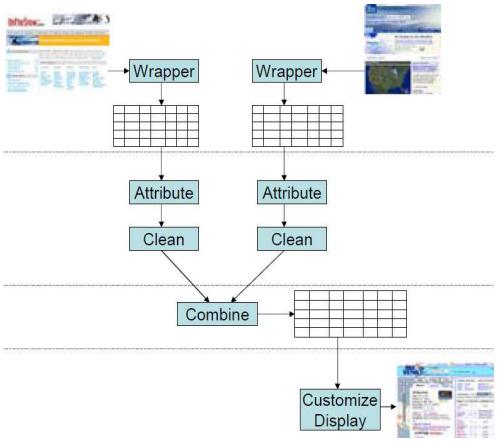
L'accès aux données

- Documents HTML vs. Bases de données
  - Génération dynamique de documents HTML
  - Formulaires web
  - Données = web caché >> web de surface
    - En 2001, 60 sites de web caché totalisaient 40 fois le web de surface
- Problèmes
  - La signification des données (identifiants, attributs) exportées sur le web se perd
  - Qualité des données, inconsistance



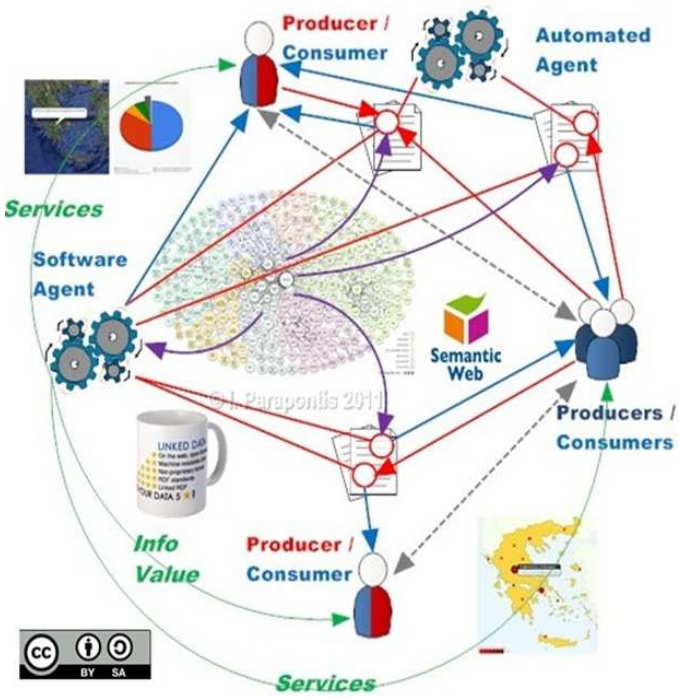
Exploitation des données du web

- Web de surface → moteurs de recherche
- Données / web caché → mashups
- Mashup
  - Intégration simple des données du web
    - « Instances » de données
    - Mots-clés, localisation
    - Union, pas de jointure
  - Approche orientée services
  - Phases d'une intégration mashup
    - Extraction des données (wrappers)
    - Calibration / nettoyage
    - Intégration
    - Affichage



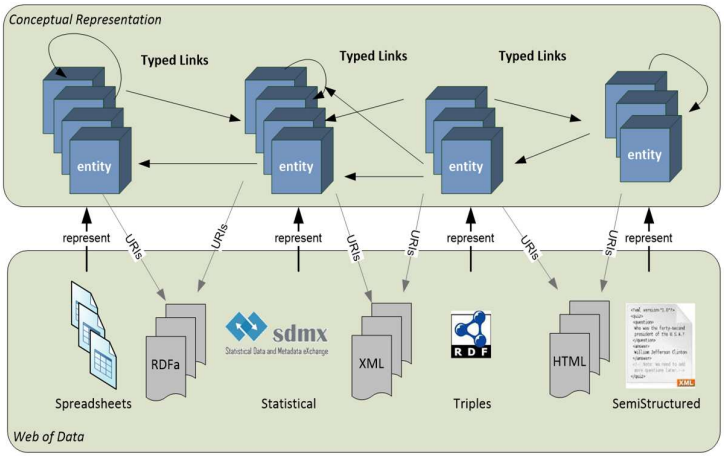
Web 3.0

- Web sémantique
  - Web 2.0 (contenus variés, producteurs / consommateurs) + sémantique
  - Vers une exploitation automatique des données du web: programmes, services, raisonnement
- Comment?
  - Le Web de Données



Web de Données

- Web d'objets décrits par des données du web
  - Descriptions d'objets
  - Liens (relations) entre ces objets
  - ***Espace de données*** global réunissant ces données

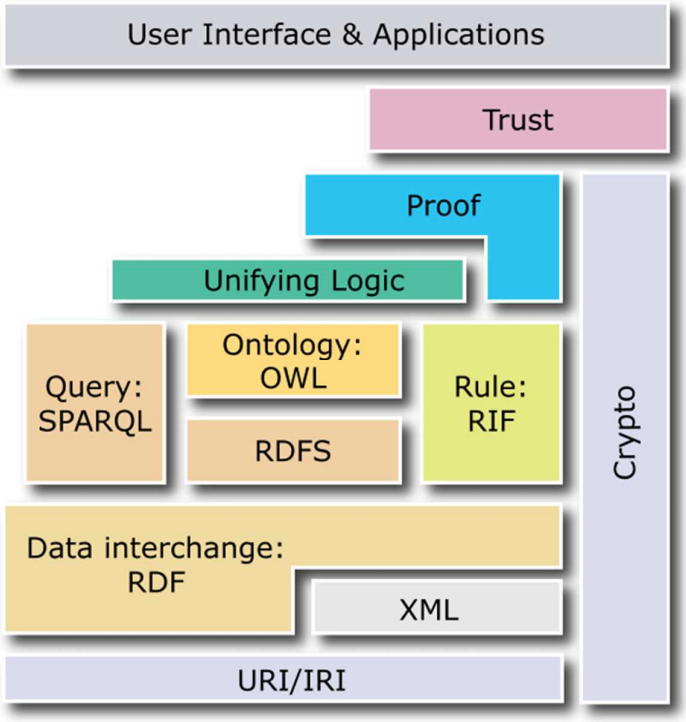


Les données ouvertes

- Données disponibles publiquement
  - Données déjà disponibles sur le web + données rendues publiques par les institutions
  - Mouvement « open data » soutenu par des initiatives gouvernementales
- Formats très divers
  - ★ Disponibles sur le web (tout format) mais avec une licence Open Data
  - ★★ En plus, format structuré (ex. excel vs. image scannée d'une table)
  - ★★★ En plus, format non propriétaire (ex. CSV au lieu d'excel)
  - ★★★★ En plus, utilisant des standards ouverts du W3C (RDF et SPARQL) pour identifier et rendre accessibles les objets par des URI déréférençables
  - ★★★★★ En plus, établissant des liens vers des objets d'autres sources

Format	Recommandation (échelle de 0 à 5)
csv	★★★
xls	★
pdf	★
doc	★
xml	★★★★
rdf	★★★★★
shp	★★★
ods	★★
tiff	★
jpeg	★
json	★★★
txt	★
html	★★

Web sémantique



RDF sur le Web

- Objectif : description sémantique des pages web
  - Web de documents → web de données → web sémantique
  - Support pour l'intégration de données à l'échelle du web
- Comment?
  - Micro-formats et micro-données
  - JSON-LD
  - RDFa
  - Linked Open Data
- Prérequis : vocabulaires communs et reconnus
  - Dublin Core, FOAF, SKOS, etc.

Quelques vocabulaires communs

- Dublin Core: description de documents/ressources
  - Contenu: *title, subject, description, source, language, relation, coverage*
  - Propriété intellectuelle: *creator, contributor, publisher, rights*
  - Autre: *date, type, format, identifier*
  - Espace de noms: http://purl.org/dc/elements/1.1/
- FOAF (Friend of a Friend): description de personnes
  - Classes (*Person, Group, Organization, Document, Image, …*)
  - Propriétés pour Person: *name, firstName, lastName, knows, homepage, …*
  - Espace de noms: http://xmlns.com/foaf/0.1/
- SKOS (Simple Knowledge Organization System): taxonomies
  - Classe *Concept*
  - Propriétés: *broader, narrower, related, prefLabel, altLabel, …*
  - Espace de noms: http://www.w3.org/2004/02/skos/core#

Micro-formats et micro-données

- Informations sémantiques rajoutées aux documents HTML
  - Utilisées par les navigateurs, les moteurs de recherche, etc.
- Micro-formats: attribut *class* faisant référence à des classes prédéfinies
  - Micro-formats prédéfinis pour personne, institution, événement, avis, etc.
  - Voir http://www.microformats.org/
  - Inconvénient: on ne peut pas décrire n'importe quel type d'objet

Ex. Utilisation du micro-format *hCard* pour décrire des personnes

```
<div class="vcard">
  <em class= "fn">Jean Dupont</em>
  <span class= "title">Ingénieur</span> chez <span
class="org">Google</span>
  <span class= "adr">
    <span class="street-address">2 rue du Moulin</span>
    <span class="locality">Village-sur-Eau</span>
    <span class="postal-code">54321</span>
  </span>
</div>
```

Micro-formats et micro-données (suite)

- Micro-données: extensible, plus puissant, on distingue types et propriétés
  - Vocabulaires prédéfinis (ex. *http://data-vocabulary.org* - Google, http://ogp.me/ns# - Open Graph pour Facebook)
  - Initiative *Schema.org* (*http://schema.org*), pour uniformiser les types de micro-données entre les principaux navigateurs

Ex. Utilisation des types *Person* et *PostalAddress*

```
<div itemscope itemtype="http://schema.org/Person">
  <span itemprop="name">Jean Dupont</span>
  <span itemprop= "jobTitle">Ingénieur</span> chez
<span itemprop="affiliation">Google</span>
  <span itemprop="address" itemscope
itemtype="http://schema.org/PostalAddress">
    <span itemprop="streetAddress">2 rue du Moulin</span>
    <span itemprop="addressLocality">Village-sur-Eau</span>
    <span itemprop="postalCode">54321</span>
  </span>
</div>
```


RDFa

- RDFa = « RDF in attributes »
  - Des descriptions RDF dans les pages (X)HTML (attributs)
  - Permet toute la richesse de RDF: URI, espaces de noms, types, …
  - Utilisées par les navigateurs spécialisés, programmes, …
    - Résultats plus riches dans les moteurs de recherche

Ex. Triplet (sujet, propriété, valeur)

```
<p about="http://monappli.monorg.com/Michel"
   property="http://xmlns.com/foaf/0.1/name">
  Michel Jordan
</p>
```

  - Attributs *about* (sujet), *property* (propriété)
  - Valeur dans le texte de la balise

RDFa (suite)

Ex. Relation entre ressources (sujet, prédicat, objet)

```
<a about="http://monappli.monorg.com/Michel"
   rel="http://purl.org/dc/elements/1.1/creator"
   href="http://www-etis.ensea.fr">
  Page créée par Michel
</a>
```

– Attributs *about* (sujet), *rel* (prédicat), *href* (objet)

Ex. Ressources objet en dehors des liens

```
<span about="http://monappli.monorg.com/ETIS"
   rel="http://monappli.monorg.com/directeur"
   resource="http://monappli.monorg.com/Mathias">
  Directeur: Mathias Quoy
</span>
```

– Attribut *resource* à la place de *href*

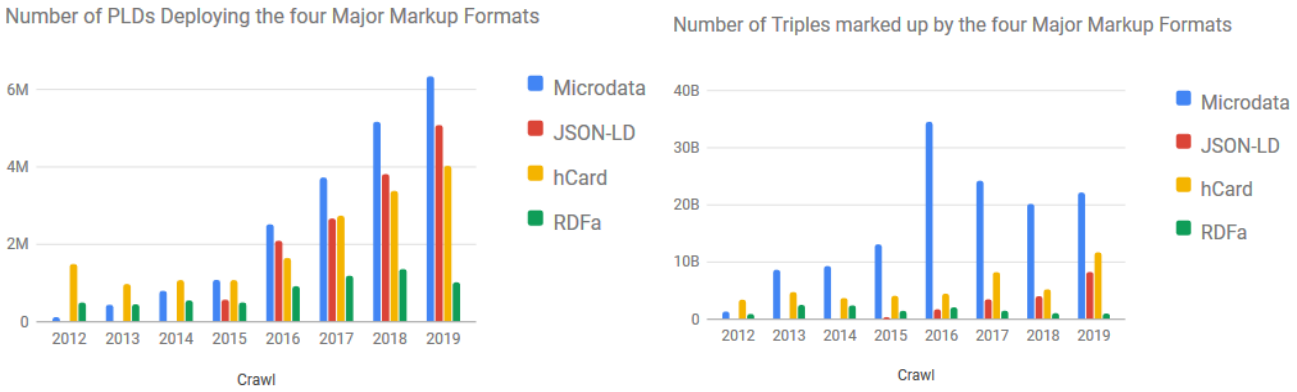
JSON-LD

- JSON for Linking Data
  - Devenu populaire avec l'utilisation massive du format JSON
  - Données en JSON en tant que scripts dans les pages HTML
- Exemple simple

```
{
  "@context": "http://monappli.monorg.com",
  "@id": "http://monappli.monorg.com/ETIS",
  "@type": "Laboratory",
  "name": "ETIS",
  "WebPage": "http://www-etis.ensea.fr",
  "member": ["http://monappli.monorg.com/Michel",
  "http://monappli.monorg.com/Dan"]
}
```

RDFa, JSON-LD, micro-données et micro-formats

- Analyse dans le temps sur un échantillon représentatif de sites
  - 2012: micro-formats, RDFa et un peu de micro-données
  - Augmentation des micro-données (effet schema.org)
  - Après 2015: JSON-LD, avec une croissance forte
  - RDFa: bon début, mais croissance faible et maintenant en baisse
- Alternative: publier des données, pas des documents annotés

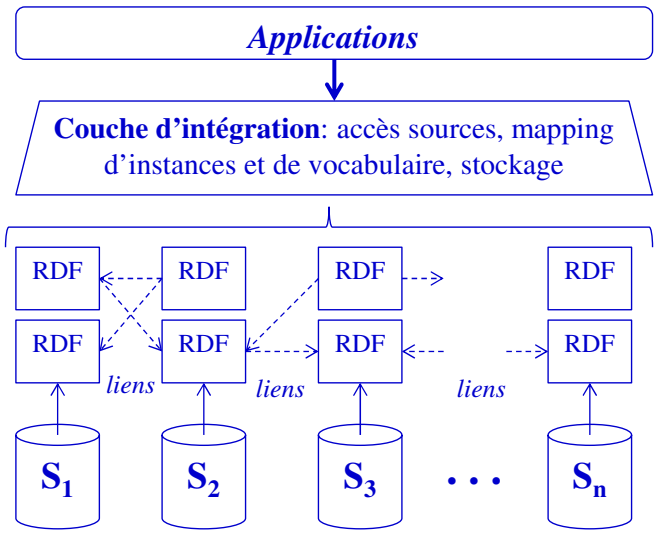


Linked Open Data (LOD)

- Espace de données global sur le Web
  - Partie ★★★★★ du Web de Données
  - Données RDF publiées par des sources différentes
  - Liens (RDF aussi) entre les données RDF de ces sources
- Les quatre principes des LOD (Tim Berners-Lee)
  - Utiliser des URI pour nommer (identifier) des choses (ressources)
  - Utiliser des URL HTTP comme URI, pour que les ressources soient accessibles sur le Web (déréférençables)
  - Quand on accède une telle URI, on retrouve quelque chose d'utile, d'informatif – en principe du RDF
  - Inclure des liens vers d'autres ressources, pour qu'on puisse découvrir de nouvelles informations

Architecture LOD

- Espace de données : trois types d'« acteurs » par rapport à chaque source
  - le publieur
  - les publieurs des autres sources
  - les consommateurs de données



Espace de données

- Architecture particulière d'intégration de données
  - Pas de schéma global défini
  - Chacun publie ses données dans son propre format et les relie aux autres
  - Amélioration progressive de la qualité du système
  - Qualité des réponses proportionnelle à l'effort d'intégration réalisé

  → adaptée à l'intégration à très large échelle et très dynamique
- En comparaison: architecture de médiateur
  - Effort important pour définir le schéma global et les mappings
  - Effort important pour maintenir le schéma et les mappings
  - Qualité garantie
- Linked Open Data: espaces de données RDF
  - Effort d'intégration: les liens entre sources
  - Deux types de liens: *d'identité des instances* et *de vocabulaire* (concepts)

Liens entre sources LOD

- Liens d'identité d'instances

```
<http://monappli.monorg.com/ETIS>
owl:sameAs
<http://hal.archives-ouvertes.fr/lab/etis> .
```

- Liens de vocabulaire

```
<http://monappli.monorg.com/Personne>
owl:equivalentClass
<http://xmlns.com/foaf/0.1/Person> .
```

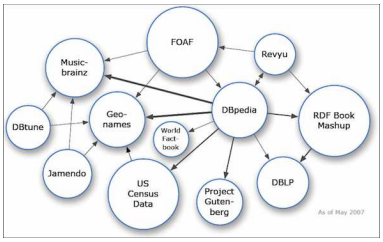
  - Types de liens: *owl:equivalentClass, owl:equivalentProperty, rdfs:subClassOf, rdfs:subPropertyOf*

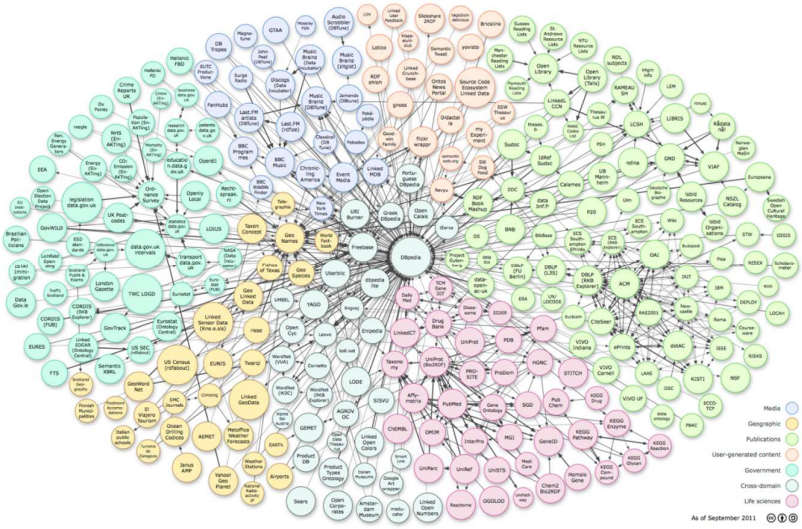
Qui fait l'effort d'intégration?

- Effort *partagé* entre publieur, autres publieurs et consommateur
  - Le publieur de la source S
    - Choisit ses vocabulaires (nouveaux ou réutilisation)
    - Publie ses données en RDF
    - Publie des liens d'identité vers d'autres sources connues
    - Publie des liens de vocabulaire vers d'autres vocabulaires connus
  - Les autres publieurs
    - Publient des liens d'identité vers les données de S
    - Publient des liens de vocabulaire vers les vocabulaires de S
  - Le consommateur = programmeur d'applications d'intégration
    - Définit le mode d'accès aux données LOD de différentes sources
    - Définit ou déduit des liens d'identité entre sources (outils spécifiques)
    - Définit ou déduit des liens entre vocabulaires (outils spécifiques)
    - Nettoie les données
    - Intègre les données (« entrepôt » RDF)
- En comparaison, avec une architecture de médiation: le consommateur fait (presque) tout

Le « nuage » LOD

- Qu'est-ce qui existe aujourd'hui sur le Web?
  - Mai 2007: 500 millions de triplets RDF, 120 000 liens RDF
  - Septembre 2011: 31,6 milliards de triplets, 503 millions liens
  - Avril 2015: le nombre de sources est multiplié par 4 compare à 2011





Sources LOD

- Quelles sources entrent dans le nuage LOD
  - Sources RDF respectant les contraintes LOD
  - Au moins 1000 triplets et 50 liens vers des sources dans le nuage
  - Accès en HTML+RDFa, ou fichier RDF, ou point d'accès SPARQL
- Qui publie des données ouvertes ?
  - Les gouvernements: UE, Etats-Unis, France (*data.gouv.fr*), …
  - Institutions culturelles: bibliothèques nationales, musées, archives
  - Autres institutions
  - Voir: https://lod-cloud.net/, http://www.w3.org/wiki/SparqlEndpoints
- Au centre du nuage LOD: ***DBpedia*** (http://dbpedia.org/)
  - Avantage DBpedia: couvre un ensemble large de concepts auxquels on peut faire référence

DBpedia

- Source LOD obtenue à partir de Wikipedia
  - Utilisation des « info boxes » sur les pages Wikipedia
  - Utilisation des catégories Wikipedia
- Version anglaise de DBpedia
  - 4,58 millions d'entités, dont 4,22 millions instances de l'ontologie DBpedia
  - 580 millions de triplets RDF
  - Autres langues: versions de DBpedia en 125 langues(!) avec liens entre les entités dans les différentes langues
    - En tout: 3 milliards de triplets RDF
- Ontologie DBpedia
  - 768 classes
  - 3000 propriétés
- Accès: SPARQL, téléchargement, autres outils

Points d'accès SPARQL

- Le mode d'accès le plus courant aux sources LOD
- Services Web de type REST
  - Plus simples que la norme SOAP
  - Accès par des requêtes HTTP GET avec paramètre *query* (ou POST)
- Ex. La requête SPARQL *q* au point d'accès *http://monsite.org/sparql*

```
GET /sparql?query=q-encodée HTTP/1.1
  Host: monsite.org
  User-agent: mon-client-sparql/0.1
```

  - La requête SPARQL *q* doit être transformée en *q-encodée* pour passer dans une URL (transformation des espaces et autres caractères interdits)
- Modalités d'appel
  - Avec un navigateur web
  - Avec une API: Javascript, PHP, Java, Python, …
  - Utilitaires en ligne de commande: *curl*, etc.

Format du résultat SPARQL

- Plusieurs formats de résultat RDF possibles
  - XML, JSON, texte (pour ASK et SELECT)
  - RDF/XML, Turtle, … (pour CONSTRUCT et DESCRIBE)
- On précise le format souhaité pour le résultat avec « Accept »

```
GET /sparql?query=q-encodée HTTP/1.1
  Host: monsite.org
  User-agent: mon-client-sparql/0.1
  Accept: application/sparql-results+xml
```

- Pour certains points d'accès: format donné par le paramètre *out*

```
GET /sparql?out=xml&query=q-encodée HTTP/1.1
  Host: monsite.org
  User-agent: mon-client-sparql/0.1
```