



Open linked data

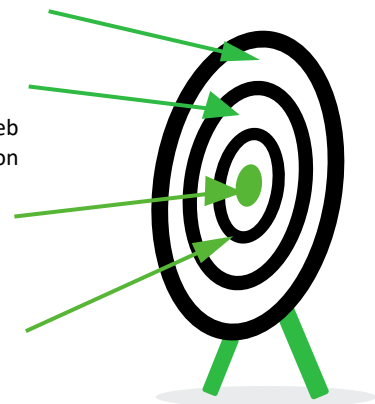
PR NAJIMA DAOUDI

1



Objectifs

- 1** Comprendre les principes fondamentaux des données liées et ouvertes, du web sémantique et de l'ontologie
- 2** Connaître les technologies et les normes des données liées, du web sémantique et de l'ontologie, ainsi que les langages de modélisation tels que RDF, OWL et RDFS
- 3** concevoir, développer et publier des ontologies et des données liées conformément aux normes du web sémantique
- 4** Utiliser des outils pour extraire et traiter des informations à partir de données liées et sémantiques pour des tâches telles que la recherche d'information, l'intégration de données et l'analyse sémantique



2

Plan



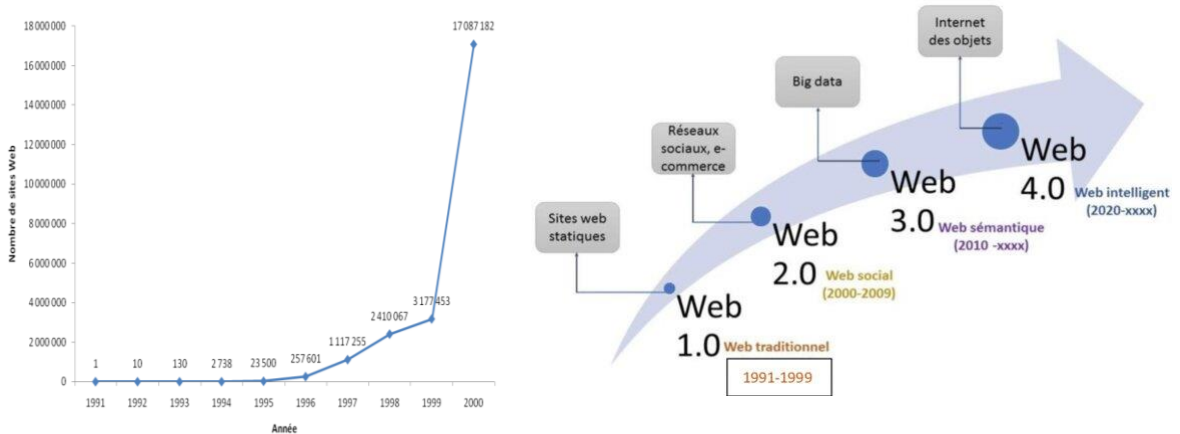
- Données ouvertes, et liées : Open linked data .
- Modèles sémantiques : RDF
- Ontologie : RDFS & OWL
- SPARQL
- Publication de données liées
- Consommer et visualiser les données liées:

LINKED OPEN DATA

3

3

Web : Croissance exponentielle et révolution



LINKED OPEN DATA

Source des données : <https://www.internetlivestats.com/total-number-of-websites/>

4

4

Recherche sur le web

- Moteur de recherche classique : fonctionnement classique !
 - Interrogation en mode similarité (requête par mot clé appelant la fourniture de documents pertinents)
 - Réponse sous forme de plusieurs liens !
 - Aucune prise en compte de la signification et le contexte des termes de recherche.
- Moteur de recherche avec des systèmes de recommandation :
 - quel degré de personnalisation!



- Besoin de **pertinence, personnalisation, de synthèse de la recherche**
 - Est-ce que les recherches classiques donnent des réponses pertinentes ?
 - Comment améliorer les recommandations pour avoir des services personnalisés?
 - comment peut on découvrir de nouvelles connaissances qu'on ne sait pas si elle existe?

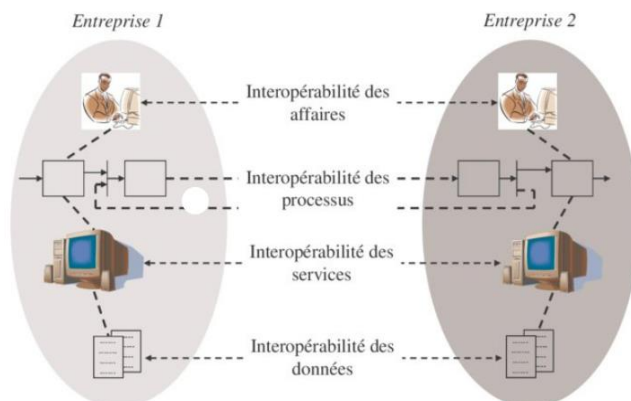
LINKED OPEN DATA

5

5

Système d'information : Intégration et Interopérabilité

- Difficulté d'accéder et de combiner des données provenant de sources diverses: prise de décisions plus compliquée.
- Difficulté d'échange et de partage des données causée par l'hétérogénéité des systèmes utilisés par différentes organisations.



LINKED OPEN DATA

6

6

Contraintes d'exploration et d'exploitation !

Hétérogénéité des données :

- les données peuvent être stockées et présentées de différentes manières, ce qui rend difficile la recherche, la comparaison et l'intégration de ces données.

Redondance des données :

- les mêmes données peuvent être stockées dans plusieurs sources différentes, ce qui peut entraîner des incohérences et des erreurs lorsqu'elles sont utilisées.

Difficulté à trouver des informations pertinentes :

- les utilisateurs peuvent avoir du mal à trouver des informations pertinentes dans des sources de données volumineuses et complexes.

Complexité des relations entre les données :

- les relations entre les données peuvent être complexes et difficiles à comprendre, ce qui peut limiter leur utilité pour les utilisateurs.

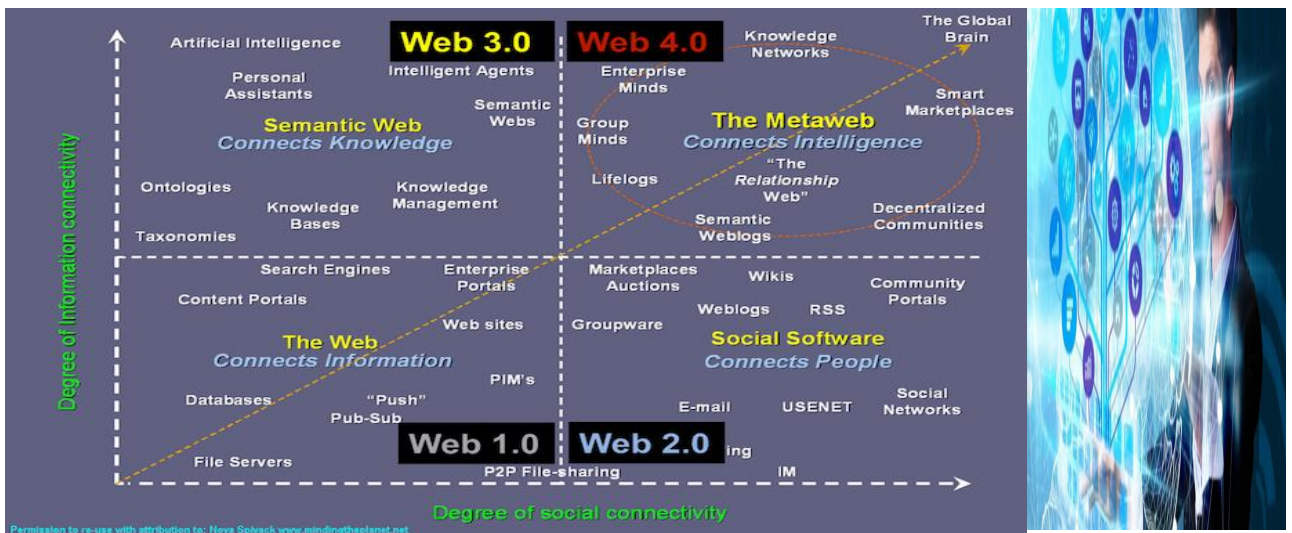
Besoin de rendre les données plus accessibles :

- les données peuvent être difficiles à trouver, à comprendre et à utiliser pour les personnes qui ne sont pas des experts du domaine.

LINKED OPEN DATA

7

7



Vers les données ouvertes et liés, des vocabulaires liés, Ontologie et web sémantique

LINKED OPEN DATA

8

8



Données ouvertes et liées : Open linked data

LINKED OPEN DATA

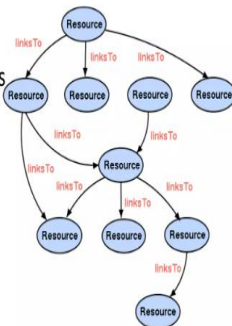
9

9

Du web des documents vers le web des données

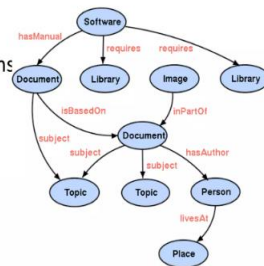
• Passer d'un web...

- Lisible par les humains
- Affichable par les machines
- « Devinable » par les machines



• ... à d'un web

- Lisible par les humains
- Affichable par les machines
- **Manipulable** par les machines



Passage d'une logique URL pour un document / à celle d'URI pour tout type de ressource

LINKED OPEN DATA

10

10

Web de données

- « We have a vision for the Web in which publishers will be encouraged to put their **data on the Web** so that it can be **accessed** and **reused** by anyone.
- **The Web of data** is a way to create a **machine-readable Web** where machines can easily **process the data and integrate it with other data to create new insights**.” Tim Berners-Lee
- Le **Web de données** est un concept qui se concentre sur la publication de **données structurées (des ressources)** sur le Web, de manière à ce qu'elles soient facilement **accessibles et interopérables**.

LINKED OPEN DATA

11

11

Web de données : Ressources

- Une **ressource** peut être définie comme un **objet abstrait** ou **physique** accessible ou non sur Internet, qui peut être **identifié** et accessible via un **URI (Uniform Resource Identifier)**.
- **Exemples** : Les ressources peuvent inclure des **personnes**, des **événements**, des **lieux**, des **concepts** et des **organisations**, ainsi que des **objets physiques** tels que des **ordinateurs**, des **capteurs** et des **véhicules**.

LINKED OPEN DATA

12

12

Web de données

Ressource: Syntaxe générique

scheme ":" ["//" authority "/"] [path] ["?" query] ["#" fragment]

- Permet de référencer une ressource.
- Le **scheme** (protocole) est **facultatif** et permet d'indiquer le protocole utilisé pour accéder à la ressource (ex : http, https, ftp, etc.).
- Le **user** (nom d'utilisateur) et **password** (mot de passe) sont **facultatifs** et permettent d'authentifier l'utilisateur pour accéder à la ressource.
- Le **host** (hôte) est **obligatoire** et représente le **nom de domaine** ou l'**adresse IP** du serveur qui héberge la ressource.
- Le **port** est **facultatif** et permet de spécifier le port d'écoute du serveur.
- Le **path** (chemin) est **obligatoire** et représente le chemin relatif ou absolu de la ressource.
- Le **query** (requête) est **facultatif** : permet de transmettre des paramètres à la ressource.
- Le **fragment** (fragment) est **facultatif** : permet d'indiquer une section particulière de la ressource.

LINKED OPEN DATA

13

13

Web de données : Ressources

scheme ":" ["//" authority "/"] [path] ["?" query] ["#" fragment]

- Toute URL est un URI, mais tout URI n'est pas forcément une URL
- Voici qlqs exemples :
- <ftp://ftp.univ-oran.dz/pub/> , <http://www.univ-oran.dz/ws.html> , <mailto:F.kateb@gmail.com> , <telnet://193.55.30.2>

Il est possible de publier ces ressources et de les lier ➔ **données liées**

LINKED OPEN DATA

14

14

Web de données liées

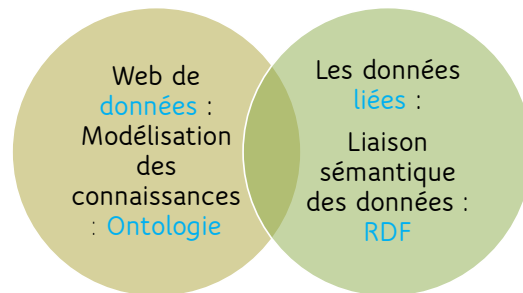
- Mise en relation des données (ressources) pour construire un réseau global qui permet à partir d'une ressource d'accéder aux autres et créer ainsi de nouvelles connaissances.
- "Le Web de données liées est une extension du Web actuel, dans laquelle les données sont liées et connectées de manière à être compréhensibles pour les ordinateurs, tout comme le sont les documents pour les humains."

Web des données liées

- Utilise les URIs pour nommer les ressources
- Utilise des URIs HTTP (URI FTP, URI SMTP ...) pour pouvoir échanger des représentations des ressources sur le Web.
- Fournit ces représentations en utilisant les standards (RDF, SPARQL)
- Inclut des liens à d'autres URIs pour permettre de découvrir de nouvelles données
- La machine sera capable de traiter la réponse du serveur : la ressource est décrite par un fichier XML (compréhensible par la machine).

Web de données liées vs Web sémantique

- le Web sémantique ajoute au web de données une signification explicite aux données à l'aide d'ontologies et de métadonnées.



Web sémantique

- « Le Web sémantique fournit un framework commun qui permet aux données d'être partagées et réutilisées entre applications, entreprises et communautés » w3c

Liaison sémantique des données : RDF

- RDF : Cadre de représentation de données
 - Permet de décrire **les propriétés et les relations** entre les ressources du Web en utilisant des triples,
 - Les triplets sont des assertions sous la forme sujet-prédicat-objet.
 - Les triples RDF sont utilisés pour décrire **les caractéristiques d'une ressource** (nom, propriétés et **ses relations** avec d'autres ressources).

Comparatif

Web actuel

- Ensemble de documents
- Basé sur HTML
- Recherche par mots-clés
- Compréhensible uniquement par l'humain

Web sémantique

- Ensemble de connaissances
- Basé sur XML/RDF
- Recherche par concepts
- Compréhensible (et donc utilisable) par la machine

Données ouvertes et liées : Linked Open Data (LOD)

- Ce sont des données publiées avec le format de données liées (respectant les règles définies par le W3C):
 - Disponibles sur le Web avec un accès ouvert
 - Disponibles sous format structuré lisible par la machine
 - Disponibles sous un format non-propriétaire
 - Identifiées par des standards ouverts du W3C
 - Liées à d'autres données

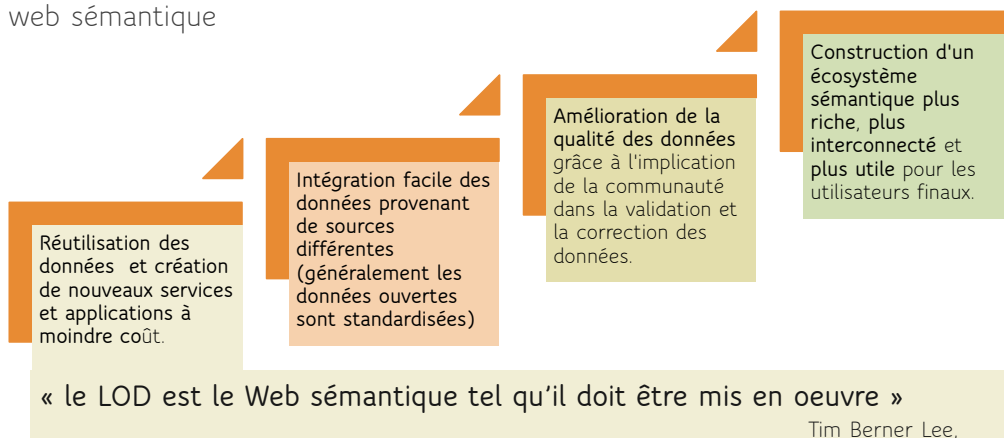
LINKED OPEN DATA

21

21

Web sémantique & Données ouvertes

- Les données ouvertes sont une ressource précieuse pour mettre en place le web sémantique —

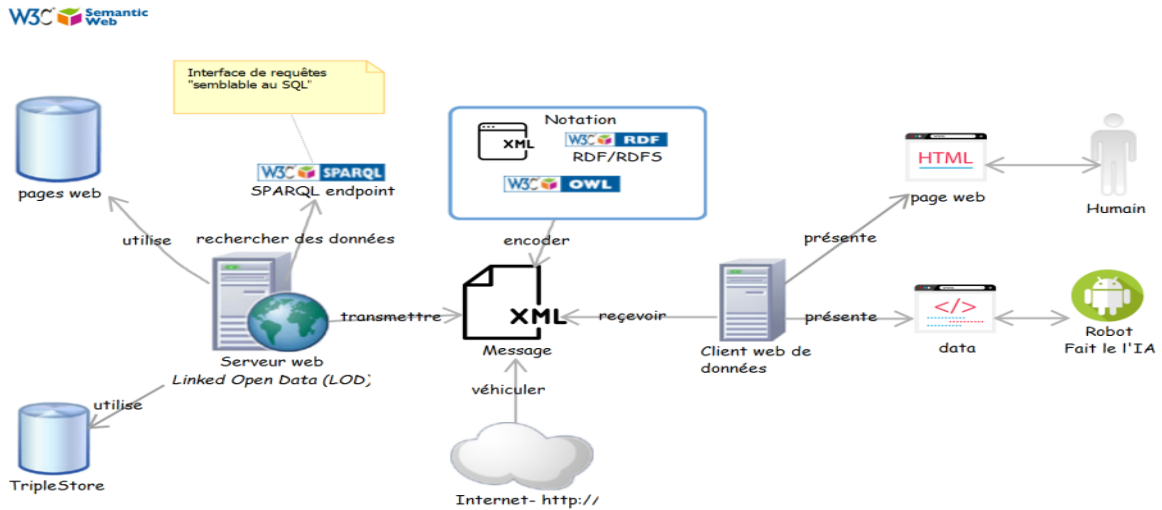


LINKED OPEN DATA

22

22

Linked open data

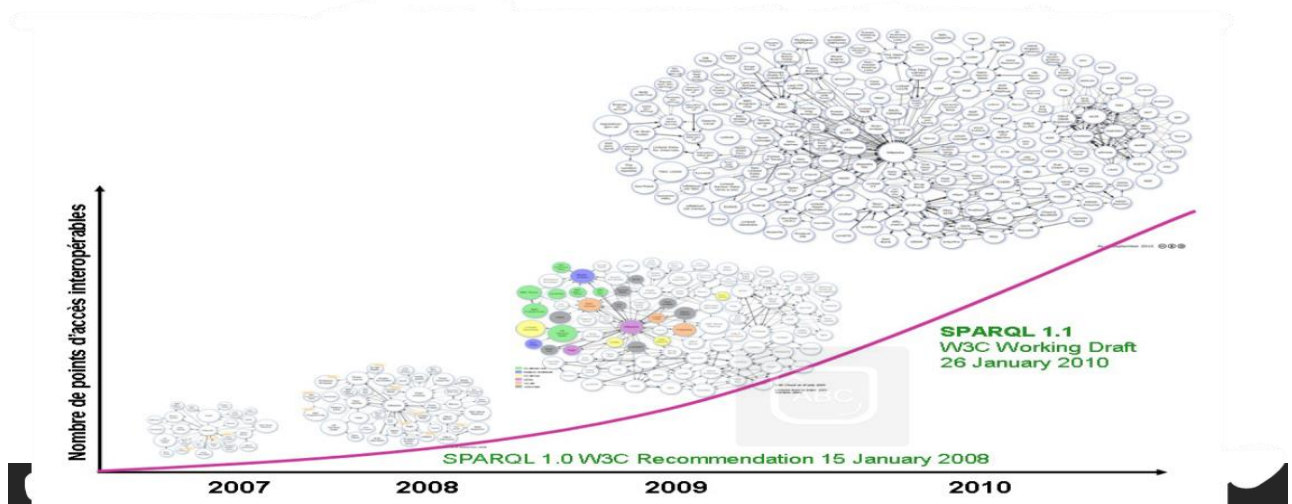


LINKED OPEN DATA

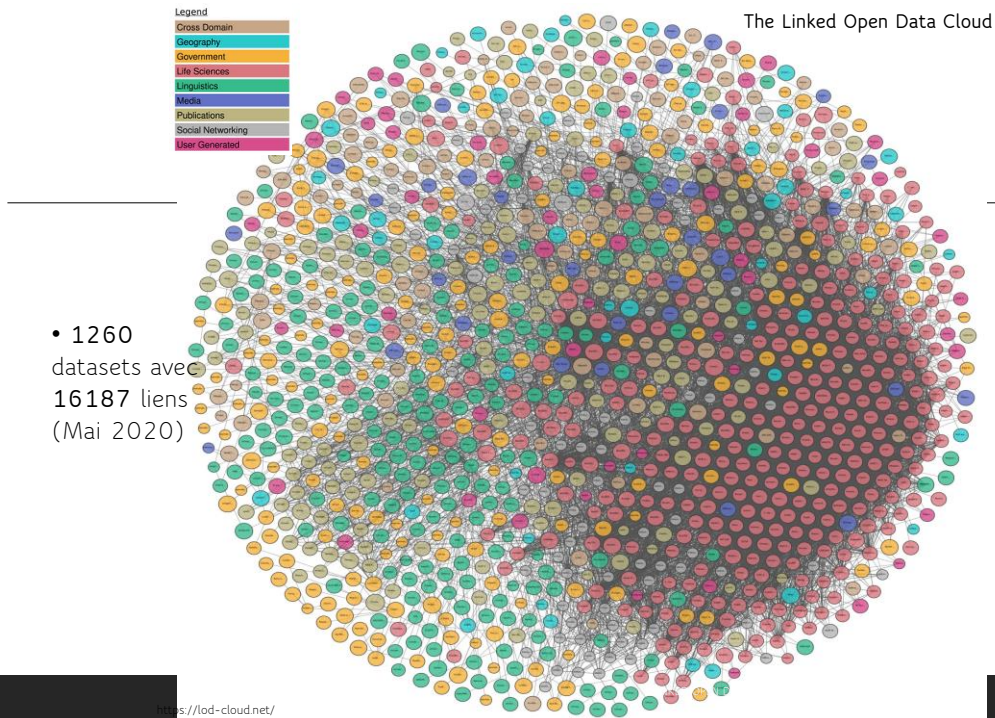
23

23

Evolution des données ouvertes et liées

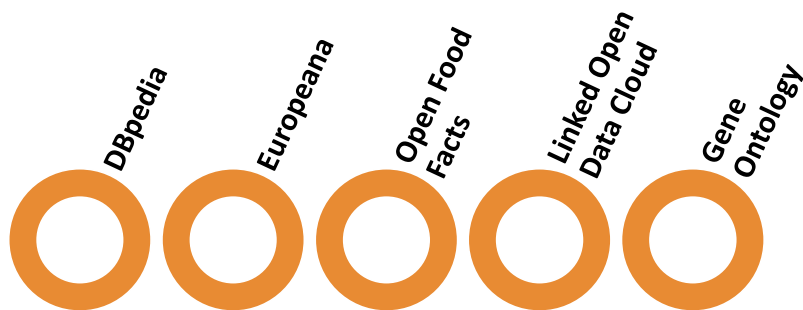


24



25

Open linked data: Découvrir quelques exemples



26

Open linked data:

Découvrir quelques exemples

- **DBpedia** : Il s'agit d'un projet qui extrait des données structurées à partir de Wikipédia et les transforme en données liées utilisant RDF. Les données liées DBpedia sont utilisées dans de nombreux domaines, tels que la recherche, les applications Web et les projets d'analyse de données.
- **Europeana** : Il s'agit d'une bibliothèque numérique en ligne qui rassemble des millions de ressources numériques provenant de bibliothèques, d'archives et de musées à travers l'Europe. Les données Europeana sont liées et utilisent des ontologies standardisées telles que EDM (Europeana Data Model) pour permettre aux utilisateurs de rechercher, explorer et interpréter les collections numériques dans un contexte plus large.

LINKED OPEN DATA

27

27

Open linked data:

Découvrir quelques exemples

- **Open Food Facts** : base de données collaborative en ligne qui collecte des informations sur les produits alimentaires du monde entier. Les données Open Food Facts sont liées et utilisent des ontologies standardisées telles que Food Ontology pour fournir des informations détaillées sur les produits alimentaires, y compris leur composition nutritionnelle et leur impact environnemental.
- **Linked Open Data Cloud** : Il s'agit d'un projet qui rassemble des milliers de sources de données ouvertes et les lie en utilisant des vocabulaires et des ontologies standardisées. Les données liées de l'Open Data Cloud sont utilisées dans de nombreux domaines, tels que la recherche, l'analyse de données, l'optimisation de la performance et les applications Web.
- **Gene Ontology** : Il s'agit d'un système de classification hiérarchique des fonctions biologiques des gènes. Les données liées de l'Ontologie des gènes sont utilisées dans la recherche en biologie, la médecine personnalisée et la découverte de médicaments pour comprendre la fonction et l'interaction des gènes dans les processus biologiques.

LINKED OPEN DATA

28

28

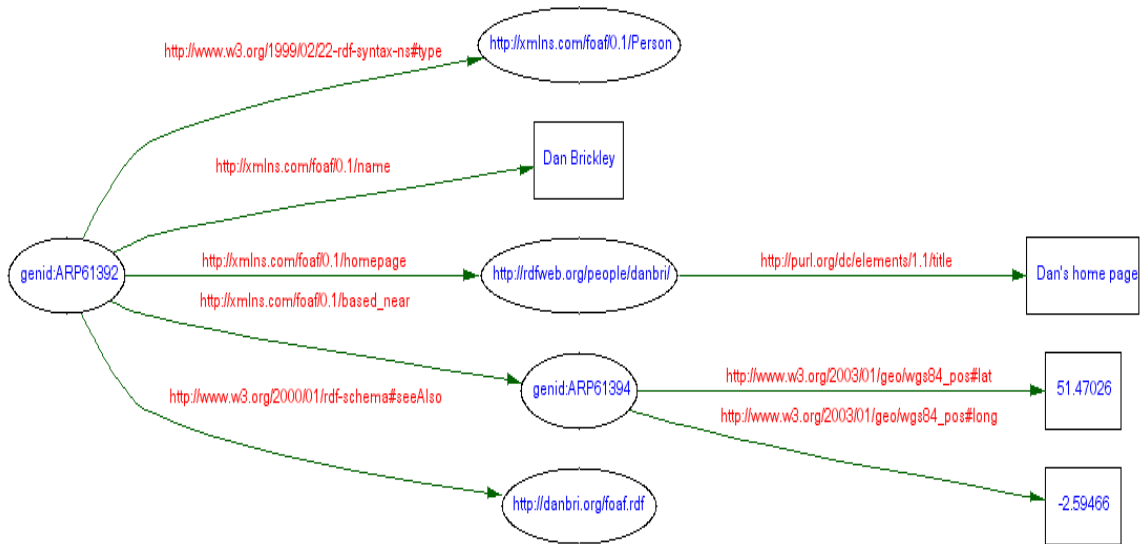
Linked Open Data Vocabularies

- Vocabulaires normalisés et partagés
- Utilisés dans les LOD
- 723 vocabulaires recensés jusqu'en octobre 2020
 - FOAF
 - GoodRelations
 - Etc,

Linked Open Data Vocabularies : FOAF (Friend Of A Friend)

- Vocabulaire RDF relatif à la description formelle des métadonnées des personnes, de leurs centres d'intérêt, de leurs relations et de leurs activités
- **Objectif**: adapter au Web sémantique le concept de « page personnelle » ou de « profil » contenant des liens vers les pages relatives aux différentes métadonnées (ville de résidence, employeur, loisirs, relations,...). Ces liens sont à leur tour ouverts et liés à d'autres.
- **Résultat**: navigation facile entre les différentes personnes

Linked Open Data Vocabularies : FOAF



LINKED OPEN DATA

<https://www.w3.org/2001/sw/Europe/talks/leon/foaf/>

31

31

Linked Open Data Vocabularies : GoodRelations

■ **Ontologie** légère pour l'échange des informations du commerce électronique, notamment des données relatives aux produits, offres, points de vente... basée sur 4 entités :

- **Agent** : personne ou organisme qui vend un Objet
- **Objet** : bien ou service fourni par un Agent
- **Offre** : Transaction proposée (vente, location, licence...)
- **Location** : Lieu où l'offre est disponible

LINKED OPEN DATA

<https://www.w3.org/wiki/GoodRelations>
<http://wiki.goodrelations-vocabulary.org/Quickstart>

32

32

Linked Open Data Vocabularies : GoodRelations

```
<gr:offers>
  <gr:Offering rdf:about="http://www.example.com/xyz#myOffering">
    <rdfs:comment xml:lang="en">We sell and repair computers and motorbikes</rdfs:comment>
    <gr:includes>
      <gr:ProductOrServicesSomeInstancesPlaceholder rdf:about="http://www.example.com/xyz#myProducts">
        <rdfs:comment xml:lang="en">computers and motorbikes</rdfs:comment>
      </gr:ProductOrServicesSomeInstancesPlaceholder>
    </gr:includes>
    <gr:hasBusinessFunction rdf:resource="http://purl.org/goodrelations/v1#Sell" />
    <gr:hasBusinessFunction rdf:resource="http://purl.org/goodrelations/v1#Repair" />
    <gr:validFrom rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2009-04-24T00:00:00+01:00</gr:validFrom>
    <gr:validThrough rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2010-04-24T00:00:00+01:00</gr:validThrough>
    <gr:availableAtOrFrom rdf:resource="http://www.example.com/xyz#myShop" />
  </gr:Offering>
</gr:offers>
```

LINKED OPEN DATA

<http://www.ebusiness-unibw.org/wiki/GoodRelationsSimple>

33

33



Couches du Web sémantique

LINKED OPEN DATA

34

34

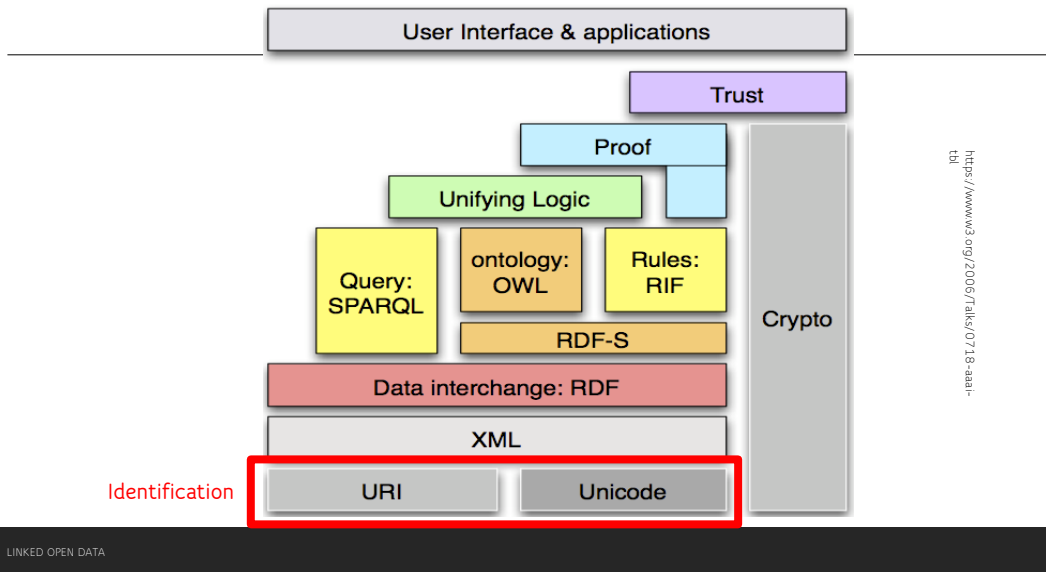
Couches du Web sémantique

- Le développement du Web sémantique se fait selon une **approche par couches (architecture en couche)**:
 - Chaque couche se construit sur celle en-dessous
 - Compatibilité descendante
 - Compréhension ascendante

Couches du Web sémantique

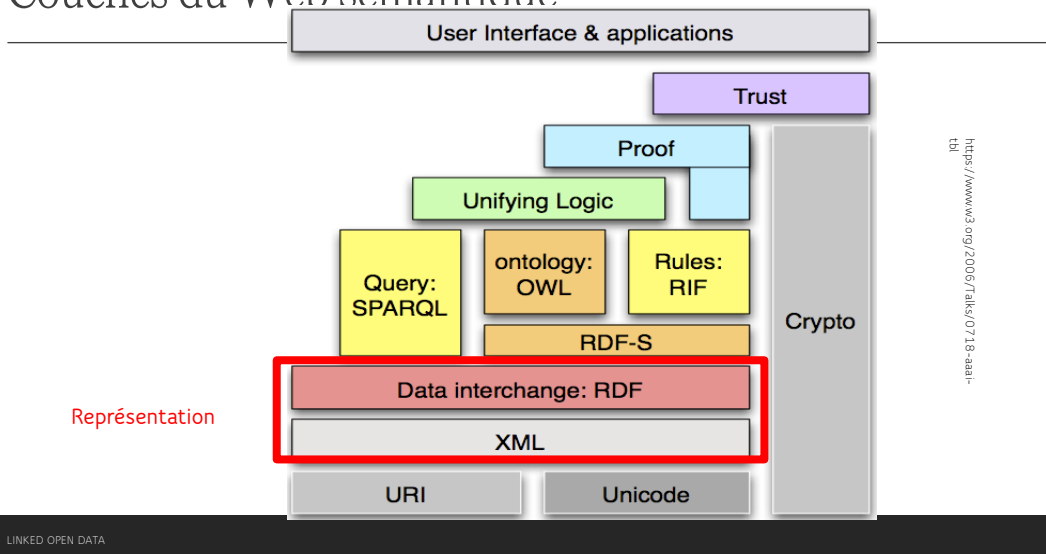
- « **Layer cake** » des standards et technologies du Web sémantique.
- Ensemble de **standards et technologies** qui permettent de **publier** des données sur le Web, de les **interroger**, de les **suivre**, et de les **tracer**.

Couches du Web sémantique



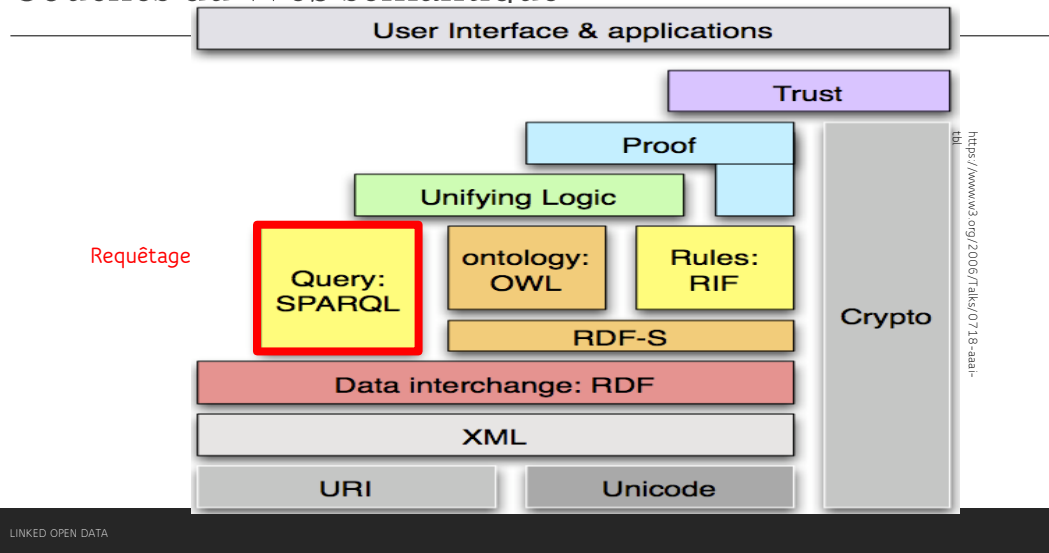
37

Couches du Web sémantique



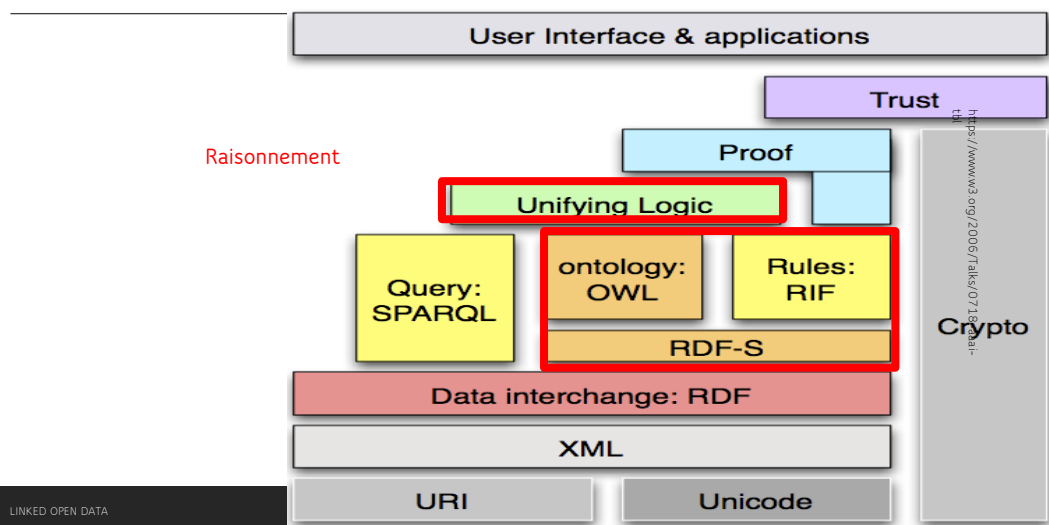
38

Couches du Web sémantique



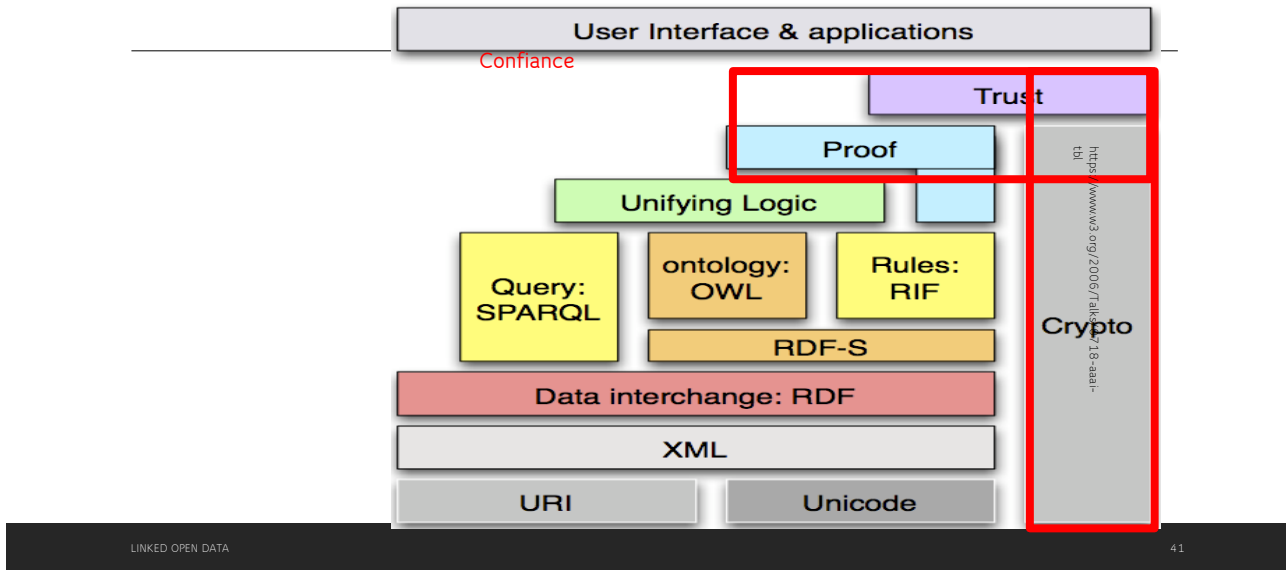
39

Couches du Web sémantique



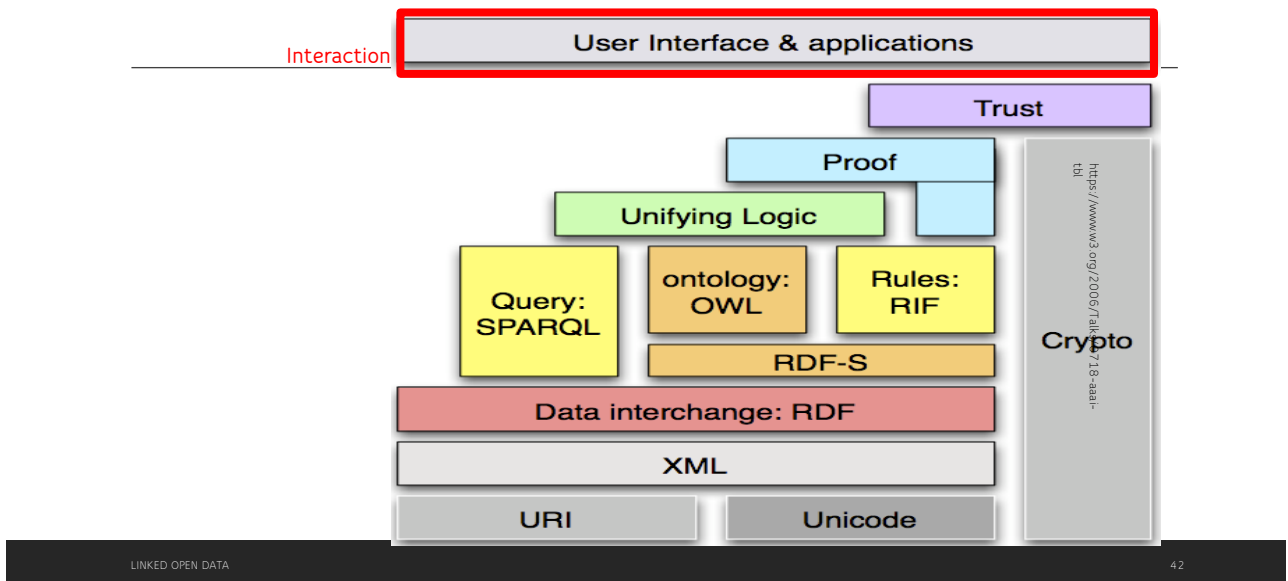
40

Couches du Web sémantique



41

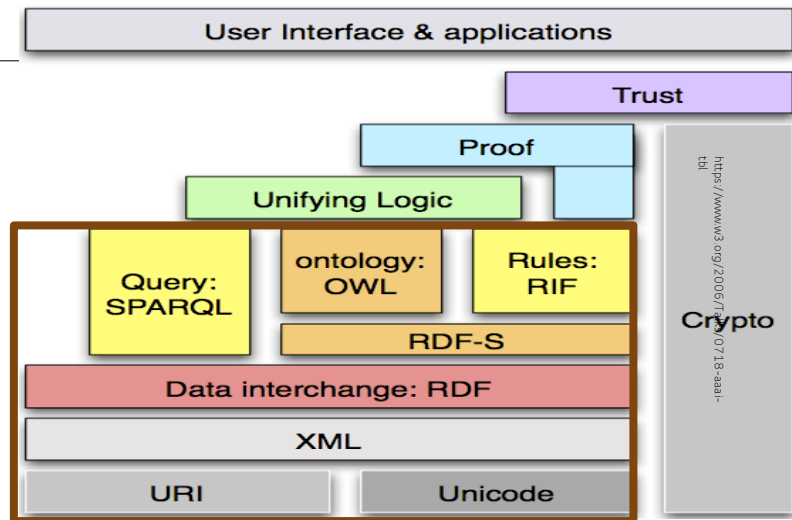
Couches du Web sémantique



42

Couches du Web sémantique

Ces couches ont des technologies associées

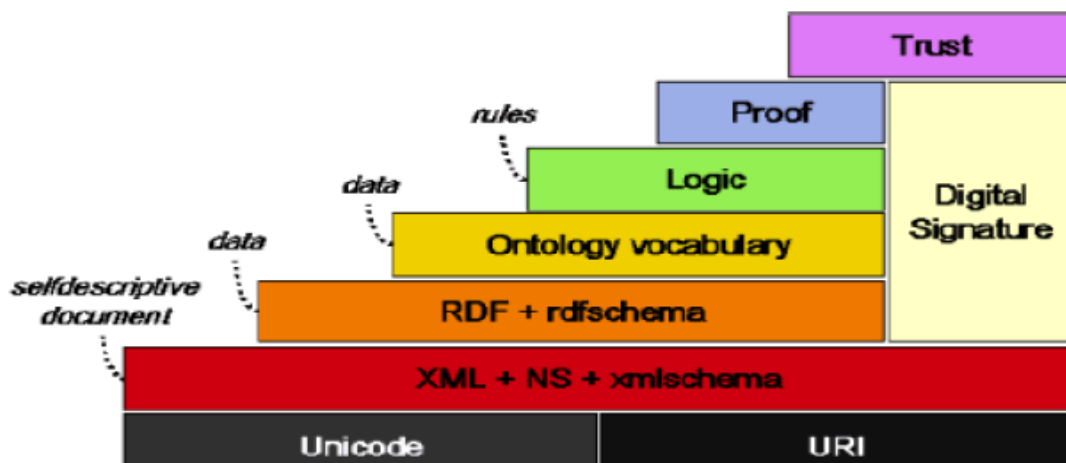


LINKED OPEN DATA

43

43

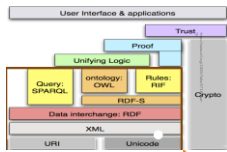
Couches du Web sémantique



LINKED OPEN DATA

44

44



Standards et technologies

XML

- Métalangage pour la représentation d'information structurée

RDF

- Resource Description Framework
- Modèle de données pour décrire les ressources du Web

SPARQL

- SPARQL Protocol and RDF Query Language
- Langage de requête sur les bases de données en RDF

RDFS

- Langage pour les ontologies

OWL

- Langage pour les ontologies
- Plus complet que RDFS

RIF

- Rules Interchange Format
- Recommandation du W3C pour standardiser l'échange de règles logiques

LINKED OPEN DATA

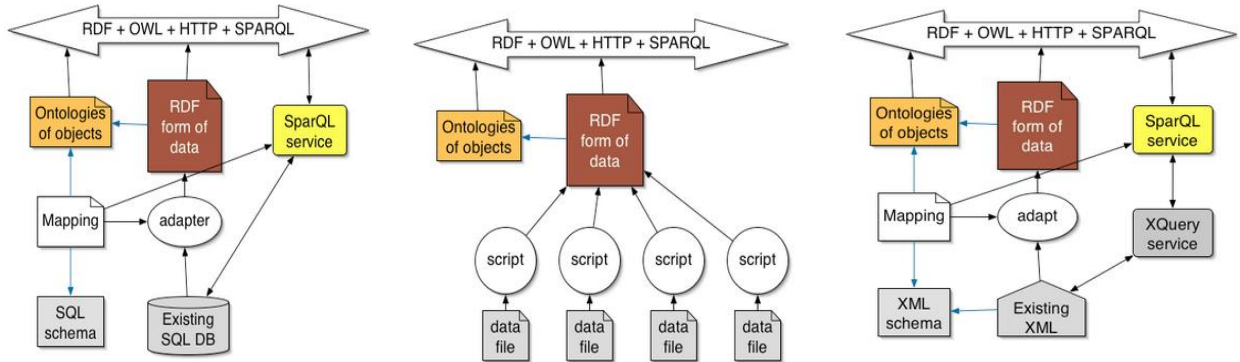
<https://www.w3.org/2005/Talks/0718-aaai->

45

45

Adaptation des systèmes existants

Les systèmes existants peuvent être adaptés au Web sémantique



LINKED OPEN DATA

<https://www.w3.org/2005/Talks/0718-aaai->

46

46



RDF

LINKED OPEN DATA

47

47

RDF : Resource Description Framework

- Standard du Web sémantique permettant de **décrire, représenter et relier des ressources**.
- **Ressource** : tout élément identifiable par un URI (page Web, bâtiment, voiture, personne, lieu...)
- **Description** : attributs des ressources; caractéristiques des ressources; relations entre les ressources
- **Framework** : cadre pour la description des ressources (langage, syntaxe...)

LINKED OPEN DATA

48

48

RDF

- RDF propose :
 - un modèle de données standardisé pour représenter les données sous forme de graphes,
 - des formats d'échanges de données normalisés tels que Turtle, N-Triples, RDF/XML, JSON-LD, etc.
- Recommandation du W3C depuis 1999
 - RDF 1.0 (2004) , RDF 1.1 (2014)

Principes de base : RDF modèle de triplets

- RDF est un modèle de triplets
 - Description des ressources par des triplets
 - (sujet, prédicat, objet)

Principes de base

RDF modèle de triplets (sujet, prédicat, objet)

■ Exemple:

- L'article "Semantic Web" a été écrit par Tim Berners-Lee, James Hendler et Ora Lassila et est paru en 2001

(Semantic Web, écrit par, Tim Berners-Lee)

(Semantic Web, écrit par, James Hendler)

(Semantic Web, écrit par, Ora Lassila)

Principes de base

RDF modèle de triplets (sujet, prédicat, objet)

■ Exemple:

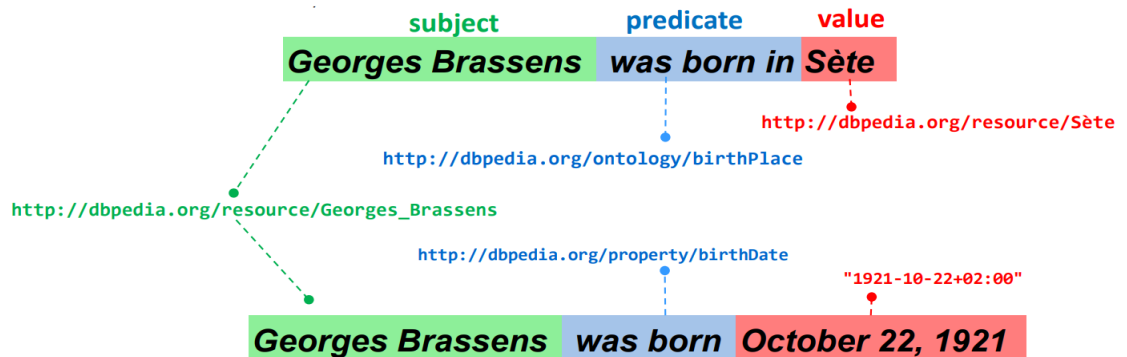
- L'article "Semantic Web" a été écrit par Tim Berners-Lee, James Hendler et Ora Lassila et est paru en 2001

- (Semantic Web, paru en, 2001)

Principes de base

RDF modèle de triplets (sujet, prédicat, objet)

Exemple.



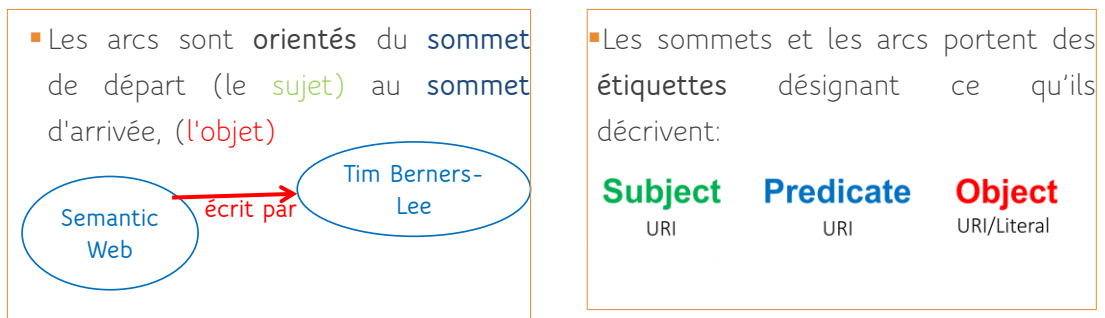
LINKED OPEN DATA

53

53

Principes de base : RDF modèle de graphes

- RDF est un modèle de graphes : (sujet, arc, sommet)
- Représentation des triplets en graphes orientés et étiquetés



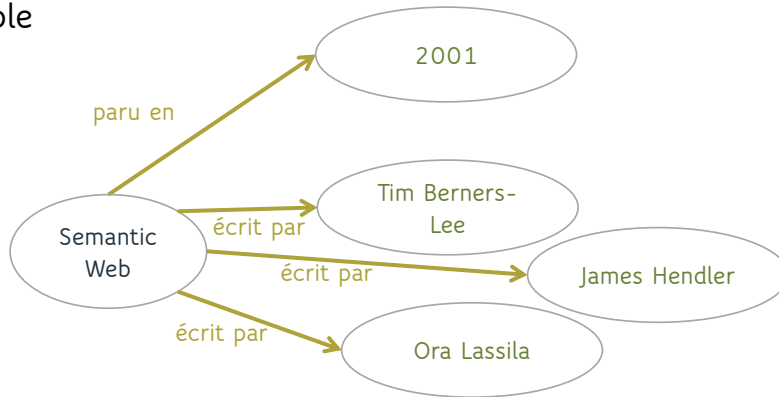
LINKED OPEN DATA

54

54

Principes de base RDF modèle de graphes (sujet, arc, sujet)

Exemple

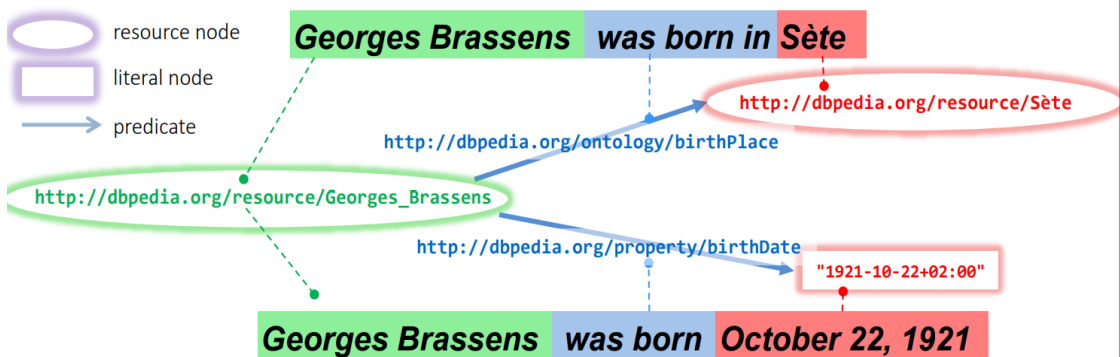


LINKED OPEN DATA

55

55

Principes de base RDF modèle de graphes (sujet, arc, sujet)



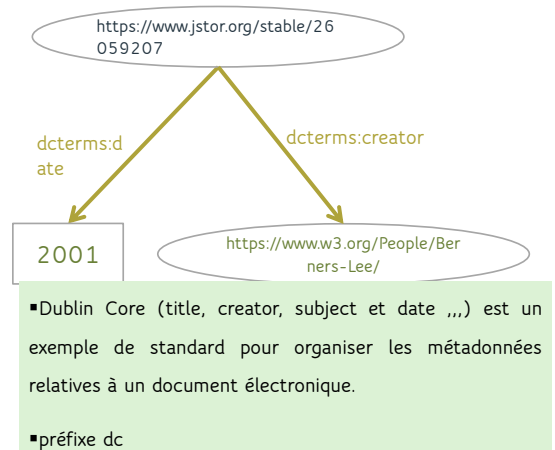
LINKED OPEN DATA

56

56

Principes de base RDF modèle de graphes (orientés & étiquetés)

- Afin d'insérer des métadonnées dans des documents, il est nécessaire de disposer d'éléments pour présenter celles-ci.
- L'utilisation de plusieurs vocabulaires au sein d'un même document risque de rendre cette tâche compliquée:
 - Les espaces de noms sont utilisés pour alléger l'écriture
 - Chaque élément ou attribut appartient à un espace de noms qui détermine le vocabulaire dont il est issu. Cette appartenance est marquée par la présence dans le nom d'un **préfixe associé à l'espace de noms**.



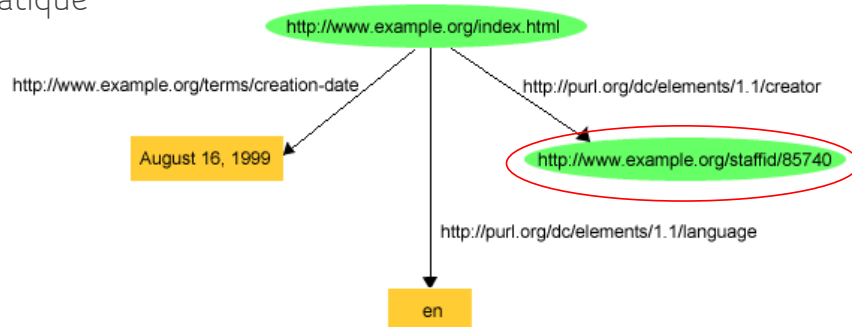
LINKED OPEN DATA

57

57

Principes de base RDF modèle de graphes (orientés & étiquetés)

Problématique



Comment représenter l'adresse (détaillée) du créateur du document?

LINKED OPEN DATA

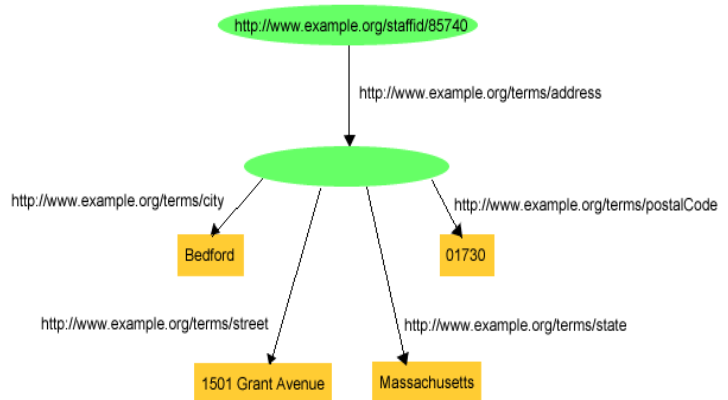
58

58

Principes de base RDF modèle de graphes (orientés & étiquetés)

■ Nœuds anonymes (Blank nodes):

- Décrire les ressources complexes
- Représenter les groupes de ressources (alternatives, listes...)
- Si un graphe comporte plusieurs nœuds anonymes, il sera nécessaire de différencier entre eux par le nom



LINKED OPEN DATA

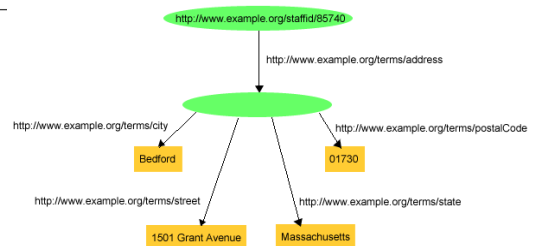
59

59

Principes de base RDF modèle de graphes (orientés & étiquetés)

■ Nœuds anonymes (Blank nodes):

Si un graphe comporte plusieurs nœuds anonymes, il sera nécessaire de différencier entre eux par le nom



| | | |
|-----------------------------------|--------------------------------|------------------|
| <code>exstaff:85740</code> | <code>exterm:address</code> | <code>???</code> |
| <code>???</code> | <code>Exterm:city</code> | |
| <code>"Bedford".</code> | | |
| <code>???</code> | <code>Exterm:street</code> | |
| <code>"1501 Grant Avenue".</code> | | |
| <code>???</code> | <code>Exterm:state</code> | |
| <code>"Massachusetts".</code> | | |
| <code>???</code> | <code>Exterm:postalCode</code> | |
| <code>"01730".</code> | | |

| | | |
|-------------------------------|--------------------------------|-------------------------|
| <code>exstaff:85740</code> | <code>exterm:address</code> | |
| <code>_:85740address</code> | | |
| <code>_:85740address</code> | <code>Exterm:city</code> | <code>"Bedford".</code> |
| <code>_:85740address</code> | <code>Exterm:street</code> | <code>"1501</code> |
| <code>Grant Avenue".</code> | | |
| <code>_:85740address</code> | <code>Exterm:state</code> | |
| <code>"Massachusetts".</code> | | |
| <code>_:85740address</code> | <code>Exterm:postalCode</code> | <code>"01730".</code> |

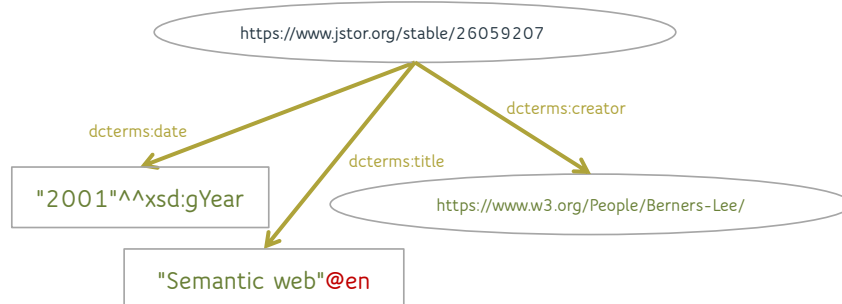
LINKED OPEN DATA

60

60

Principes de base RDF modèle de graphes (orientés & étiquetés)

- Possibilité de spécifier la langue d'un littéral textuel selon les valeurs possibles de l'attribut `xml:lang`



LINKED OPEN DATA

61

61

Principes de base RDF modèle de graphes (orientés & étiquetés)

- Typage des valeurs littérales
 - Par défaut, une valeur littérale est une chaîne de caractères de type `xsd:string`
 - Utilisation de certains types définis par `XSD` pour spécifier le type d'une valeur littérale :
`"valeur"^^<http://www.w3.org/2001/XMLSchema#type>` ou `"valeur"^^xsd:type`

- `xsd:string` : chaînes de caractères.
- `xsd:integer` : nombres entiers.
- `xsd:boolean` : valeurs booléennes
- `xsd:date` : dates.
- `xsd:dateTime` : dates et heures.

LINKED OPEN DATA

(*). <https://www.w3.org/TR/rdf-nt/#DTYPEINTERP>

62

62

Validation de documents RDF

- <https://www.w3.org/RDF/Validator/rdfval> : outil en ligne proposé par W3C afin de valider des documents RDF (RDF/XML, Turtle, N-Triples, etc.) en vérifiant leur conformité aux spécifications du langage RDF. Voici quelques cas d'utilisation pour cet outil :
 - **Validation syntaxique** : valider la syntaxe des documents RDF et s'assurer qu'ils sont conformes aux spécifications du langage RDF.
 - **Validation sémantique** : valider la sémantique de vos données RDF en vérifiant que les termes utilisés dans le document RDF correspondent à des ontologies ou des vocabulaires standardisés.
 - **Débogage** : En cas d'erreur lors de l'utilisation de vos données RDF dans une application ou un outil, le RDF Validator peut être utilisé pour identifier la source du problème en vérifiant la syntaxe et la sémantique de votre document RDF.

Sérialisation RDF

Sérialisation

- Afin de permettre une représentation des données RDF de manière standardisée et interopérable, il existe plusieurs syntaxes concrètes pour écrire des graphes RDF dans un format textuel : Sérialisation.



LINKED OPEN DATA

65

65

Sérialisation

Il existe de nombreux formats de sérialisation :

Les plus connues :

- **RDF/XML**
 - norme RDF/XML : syntaxe XML pour représenter un graphe RDF
 - élément **Description** pour décrire une ressource attribut **about** pour le sujet,
 - sous-élément pour la propriété, contenu du sous-élément pour la propriété (qui peut être parfois simplifié en attribut)
 - on peut regrouper dans un même élément Description toutes les propriétés dont cette ressource est sujet.
 - *Difficile à lire par un humain, réservé à la machine*
- **N-Triples**
 - sérialisation sous forme de triplets
 - chaque triplet est écrit sous la forme :
 - <IRI du sujet> <IRI du prédicat> <IRI de l'objet ou littéral>
 - Ex : <http://example.org/bob#me> <http://xmlns.com/foaf/0.1/knows> <http://example.org/alice#me>
 - *Plus facile à lire par un humain,*
- **Turtle (Terse RDF Triple Language)**
 - dérivé de N-Triples, plus concis, avec des facilités syntaxiques pour rendre le code plus lisible
 - *Plus concis, plus facile encore à lire par un humain*
- **TriG**
 - Extension de Turtle pour utiliser plusieurs graphes

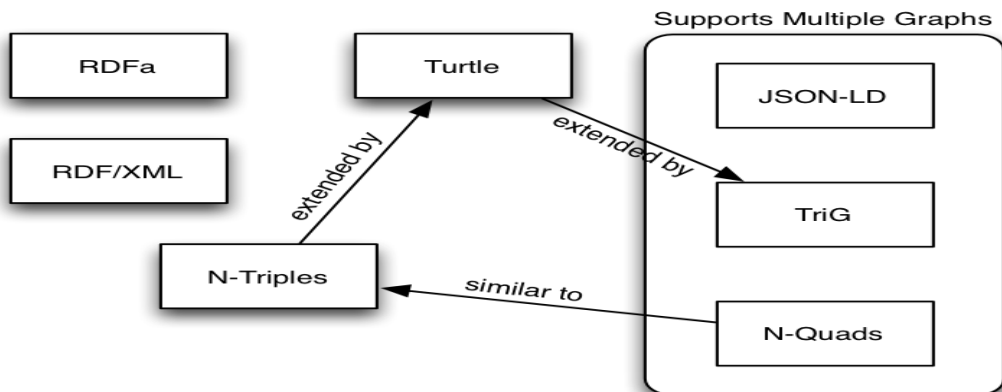
LINKED OPEN DATA

66

66

Formats de sérialisation

RDF 1.1



LINKED OPEN DATA

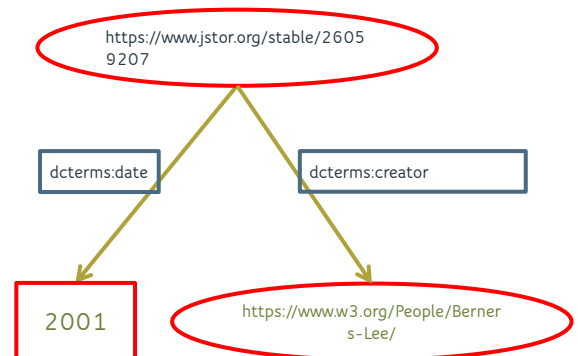
<https://www.w3.org/TR/rdf11/>

67

67

Sérialisation : RDF/XML

- Première syntaxe de RDF
- Représente les nœuds du graphe en éléments **nœuds**
- Représente les arcs du graphe en éléments **propriétés**



LINKED OPEN DATA

68

68

RDF/XML

Racine

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdfsyntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms#"
  <rdf:Description rdf:about="https://www.jstor.org/stable/26059207">
    <dcterms:date>2001</dcterms:date>
    <dcterms:creator>
      <rdf:Description rdf:about="https://www.w3.org/People/Berners-Lee/">
      </rdf:Description>
    </dcterms:creator>
  </rdf:Description>
</rdf:RDF>
```

LINKED OPEN DATA

69

69

RDF/XML

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdfsyntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms#"
  <rdf:Description rdf:about="https://www.jstor.org/stable/26059207">
    <dcterms:date>2001</dcterms:date>
    <dcterms:creator>
      <rdf:Description rdf:about="https://www.w3.org/People/Berners-Lee/">
      </rdf:Description>
    </dcterms:creator>
  </rdf:Description>
</rdf:RDF>
```

Description des ressources

LINKED OPEN DATA

70

70

RDF/XML

URI du
Sujet

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdbsyntax-ns#"
          xmlns:dcterms="http://purl.org/dc/terms#">
  <rdf:Description rdf:about="https://www.jstor.org/stable/26059207">
    <dcterms:date>2001</dcterms:date>
    <dcterms:creator>
      <rdf:Description rdf:about="https://www.w3.org/People/Berners-Lee/">
        </rdf:Description>
      </dcterms:creator>
    </rdf:Description>
  </rdf:RDF>
```

LINKED OPEN DATA

71

71

RDF/XML

Prédicats

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdbsyntax-ns#"
          xmlns:dcterms="http://purl.org/dc/terms#">
  <rdf:Description rdf:about="https://www.jstor.org/stable/26059207">
    <dcterms:date>2001</dcterms:date>
    <dcterms:creator>
      <rdf:Description rdf:about="https://www.w3.org/People/Berners-Lee/">
        </rdf:Description>
      </dcterms:creator>
    </rdf:Description>
  </rdf:RDF>
```

LINKED OPEN DATA

72

72

RDF/XML

URI de l'Objet

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdbsyntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms#"
  <rdf:Description rdf:about="https://www.jstor.org/stable/26059207">
    <dcterms:date>2001</dcterms:date>
    <dcterms:creator>
      <rdf:Description rdf:about="https://www.w3.org/People/Berners-Lee/">
    </rdf:Description>
    </dcterms:creator>
  </rdf:Description>
</rdf:RDF>
```

LINKED OPEN DATA

73

73

RDF/XML

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdbsyntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms#"
  <rdf:Description rdf:about="https://www.jstor.org/stable/26059207">
    <dcterms:date>2001</dcterms:date> Valeur littérale
    <dcterms:creator>
      <rdf:Description rdf:about="https://www.w3.org/People/Berners-Lee/">
    </rdf:Description>
    </dcterms:creator>
  </rdf:Description>
</rdf:RDF>
```

LINKED OPEN DATA

74

74

RDF/XML

- Lorsqu'un objet dans un triplet RDF n'a pas d'autres arcs sortants, il est possible d'utiliser l'attribut "rdf:resource" dans l'élément de propriété représentant le prédicat pour alléger la syntaxe.
- Cela permet d'éviter la redondance de l'objet dans le triplet RDF et rend la représentation des données plus concise.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="...">
  <rdf:Description rdf:about="sujet">
    <prédictat rdf:resource="objet"/>
    <prédictat>objet littéral</prédictat>
  </rdf:Description>
</rdf:RDF>
```

LINKED OPEN DATA

75

75

Syntaxe : RDF/XML

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdfsyntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms#">
  <rdf:Description rdf:about="https://www.jstor.org/stable/26059207">
    <dcterms:date>2001</dcterms:date>
    <dcterms:creator>
      <rdf:Description rdf:about="https://www.w3.org/People/Berners-Lee/">
    </rdf:Description>
    </dcterms:creator>
  </rdf:Description>
</rdf:RDF>
```

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdfsyntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms#">
  <rdf:Description rdf:about="https://www.jstor.org/stable/26059207">
    <dcterms:date>2001</dcterms:date>
    <dcterms:creator rdf:resource="https://www.w3.org/People/Berners-Lee/">
  </rdf:Description>
</rdf:RDF>
```

LINKED OPEN DATA

76

76

RDF/XML

- Si un objet a une valeur littérale, il peut être représenté comme attribut XML de l'élément du sujet correspondant

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms#">
  <rdf:Description rdf:about="https://www.jstor.org/stable/26059207">
    <dcterms:date>2001</dcterms:date>
    <dcterms:creator rdf:resource="https://www.w3.org/People/Berners-Lee/" />
  </rdf:Description>
</rdf:RDF>
```



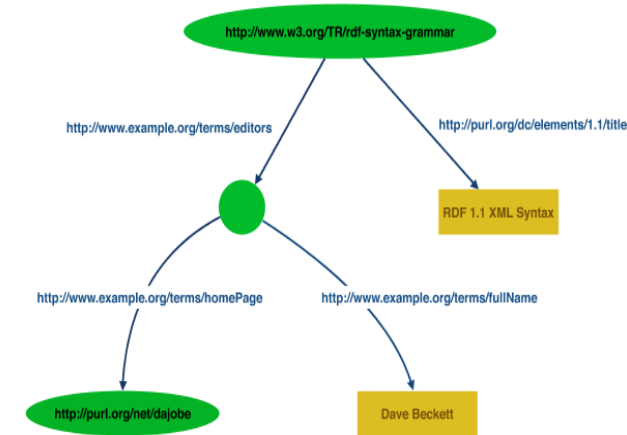
```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms#">
  <rdf:Description rdf:about="https://www.jstor.org/stable/26059207"
    dcterms:date="2001" >
    <dcterms:creator rdf:resource="https://www.w3.org/People/Berners-Lee/" />
  </rdf:Description>
</rdf:RDF>
```

Syntaxe : RDF/XML

- Spécification du type d'une valeur littérale avec l'attribut `rdf:datatype`
- Spécification de la langue d'une valeur littérale textuelle avec l'attribut `xml:lang`

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/terms">
  <rdf:Description rdf:about="https://www.jstor.org/stable/26059207">
    <dcterms:date rdf:datatype="http://www.w3.org/2001/XMLSchema#Year">2001</dcterms:date>
    <dcterms:title xml:lang="en">Semantic web</dcterms:title>
    <dcterms:creator rdf:resource="https://www.w3.org/People/Berners-Lee/" />
  </rdf:Description>
</rdf:RDF>
```

RDF/XML



LINKED OPEN DATA

- Identification des nœuds anonymes avec l'attribut `rdf:nodeID`

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:exterms="http://www.example.org/terms/"
  xmlns:dc="http://purl.org/dc/elements/1.1">
  <rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar"
    dc:title="RDF 1.1 XML Syntax">
    <exterms:editors rdf:nodeID="_:editors"/>
  </rdf:Description>

  <rdf:Description rdf:nodeID="_:editors" exterms:fullName="Dave Beckett">
    <exterms:homePage rdf:resource="http://purl.org/net/dajobe"/>
  </rdf:Description>
</rdf:RDF>
```

79

RDF/XML : Gestion de groupes

- Gestion de groupes de ressources ou de valeurs littérales avec les attributs permet de :
 - Représenter des relations complexes entre des données RDF.
 - Ensemble de ressources pourrait représenter une collection de livres d'une bibliothèque.
 - Minimiser la redondance d'informations dans les données RDF.
 - groupe de ressources pour représenter plusieurs auteurs d'un livre,
 - Intégrer de contraintes ou de règles,
 - groupe ordonné de ressources pour représenter une séquence d'étapes dans un processus métier (étape : ressource unique).



rdf:Bag

pour un
groupe non
ordonné

rdf:Seq

pour un
groupe
ordonné

rdf:Alt

pour un
groupe
d'alternatives

rdf:List

pour les listes
exhaustives
et ordonnées

LINKED OPEN DATA

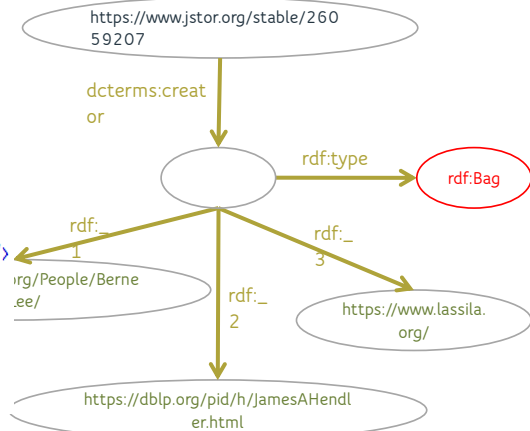
80

80

RDF/XML

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms">

  <rdf:Description rdf:about="https://www.jstor.org/stable/26059207">
    <dcterms:creator>
      <rdf:Bag>
        <rdf:li rdf:resource="https://www.w3.org/People/Berners-Lee/">
        <rdf:li rdf:resource="https://dblp.org/pid/h/JamesAHendler.html"/>
        <rdf:li rdf:resource="https://www.lassila.org/">
      </rdf:Bag>
    </dcterms:creator>
  </rdf:Description>
</rdf:RDF>
```



LINKED OPEN DATA

81

81

Syntaxe : RDF/XML

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms">

  <rdf:Description rdf:about="https://www.jstor.org/stable/26059207">
    <dcterms:creator>
      <rdf:Seq>
        <rdf:li rdf:resource="https://www.w3.org/People/Berners-Lee/">
        <rdf:li rdf:resource="https://dblp.org/pid/h/JamesAHendler.html"/>
        <rdf:li rdf:resource="https://www.lassila.org/">
      </rdf:Seq>
    </dcterms:creator>
  </rdf:Description>
</rdf:RDF>
```

LINKED OPEN DATA

82

82

RDF/XML

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms">

  <rdf:Description rdf:about="https://www.jstor.org/stable/26059207">
    <dcterms:title>
      <rdf:Alt>
        <rdf:li xml:lang="en">Semantic web</rdf:li>
        <rdf:li xml:lang="fr">Web sémantique</rdf:li>
        <rdf:li xml:lang="ar">الويب الدلالي</rdf:li>
      </rdf:Alt>
    </dcterms:title>
  </rdf:Description>
</rdf:RDF>
```

LINKED OPEN DATA

83

83

RDF/XML

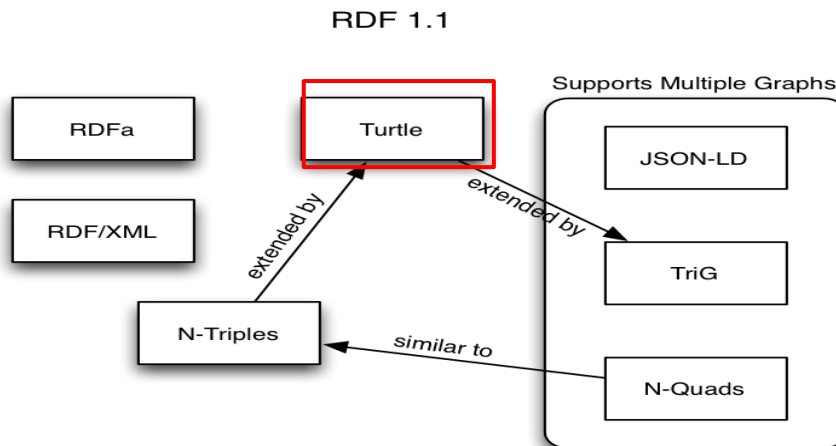
- <https://www.w3.org/TR/rdf-syntax-grammar/>

LINKED OPEN DATA

84

84

Formats de sérialisation



LINKED OPEN DATA

<https://www.w3.org/TR/rdf11->

85

85

Turtle (Terse RDF Triple Language)

- Introduit l'utilisation d'abréviations pour les éléments les plus utilisés
 - Les URIs sont mis entre < >
 - Les valeurs littérales sont mises entre " "
 - Les triplets sont séparés par .
 - Typage des valeurs littérales par "valeur"^^xsd:type
 - Spécification de la langue avec "valeur"@lang
 - Nœuds anonymes _:nom

LINKED OPEN DATA

86

86

Turtle

- Les prédicats d'un même sujet sont séparés par ;
`<https://www.jstor.org/stable/26059207> <http://purl.org/dc/terms/date> "2001";`
`<http://purl.org/dc/terms/creator> <https://www.w3.org/People/Berners-Lee/> .`
- Les objets introduits par le même prédicat sont séparés par ,
`<https://www.jstor.org/stable/26059207> <http://purl.org/dc/terms/date>`
`"2001";`
`<http://purl.org/dc/terms/creator> <https://www.w3.org/People/Berners-Lee/> ,`
`<https://dblp.org/pid/h/JamesAHendler.html> , <https://www.lassila.org/> .`

Turtle

- Les espaces de noms sont introduits par @prefix

@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 @prefix dcterms:<http://purl.org/dc/terms#>

`<https://www.jstor.org/stable/26059207> dcterms:date "2001";`
`dcterms:creator <https://www.w3.org/People/Berners-Lee/> ,`
`<https://dblp.org/pid/h/JamesAHendler.html> , <https://www.lassila.org/> .`

Quelques préfix commun

| prefix... | ...stands for |
|-----------|---------------------------------------|
| rdf: | http://xmlns.com/foaf/0.1/ |
| rdfs: | http://www.w3.org/2000/01/rdf-schema# |
| owl: | http://www.w3.org/2002/07/owl# |
| xsd: | http://www.w3.org/2001/XMLSchema# |
| dc: | http://purl.org/dc/elements/1.1/ |
| foaf: | http://xmlns.com/foaf/0.1/ |

LINKED OPEN DATA

89

89

Turtle



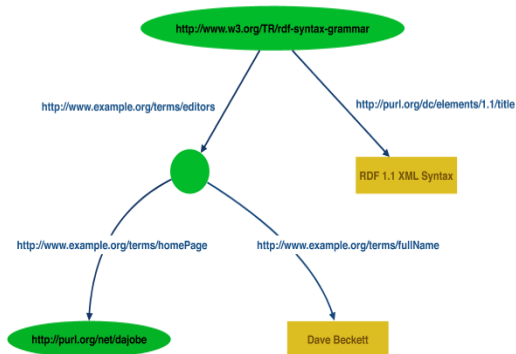
LINKED OPEN DATA

90

90

Turtle

- Les nœuds anonymes sont mis entre []



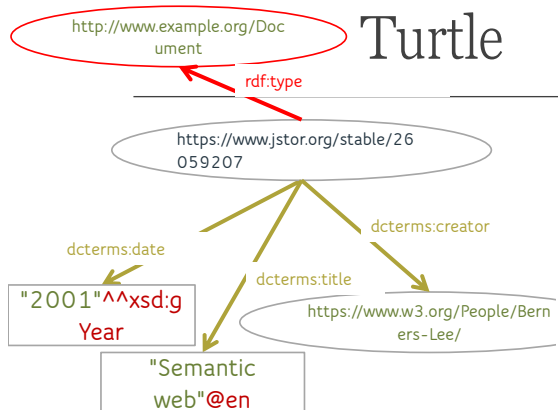
```
@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix dc:<http://purl.org/dc/elements/1.1/#>
@prefix exterms:<http://www.example.org/terms #>
```

```
<http://www.w3.org/TR/rdf-syntax-grammar> dc:title "RDF 1.1 XML
Syntax";
exterms:editors [exterms:homePage <http://purl.org/net/dajobe> ;
exterms:fullName "Dave Beckett" ].
```

LINKED OPEN DATA

91

91



- Typage des ressources avec l'attribut a

```
@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix dcterms:<http://purl.org/dc/terms/#>
@prefix ex:<http://www.example.org/ #>
```

```
<https://www.jstor.org/stable/26059207> a ex:Document ;
dcterms:date "2001"^^xsd:gYear ; dcterms:title "Semantic
web"@en; dcterms:creator <https://www.w3.org/People/Berners-
Lee/>.
```

LINKED OPEN DATA

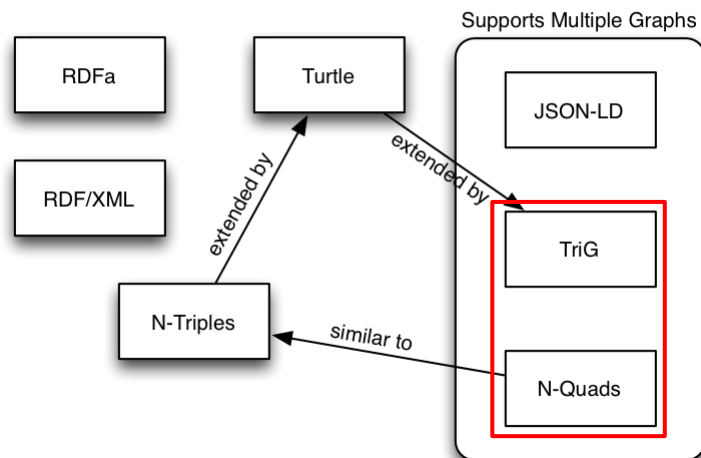
92

92

Turtle

■ <https://www.w3.org/TR/turtle/>

Formats de sérialisation



Syntaxe

- TriG et N-Quads permettent d'exprimer des contextes, i.e de nommer des graphes → nécessaire quand on a un grand graphe composé de plusieurs graphes (cas du GGG)

Exemple TriG

```
@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
@prefix dcterms:<http://purl.org/dc/terms/#>
```

```
GRAPH <http://www.example.org/Articles>
```

```
{
  <https://www.jstor.org/stable/26059207> dcterms:creator
  ( <https://www.w3.org/People/Berners-Lee/> <https://dblp.org/pid/h/JamesAHendler.html/>
    <https://www.lassila.org/> ) .
}
```

```
GRAPH <http://www.example.org/Livres>
```

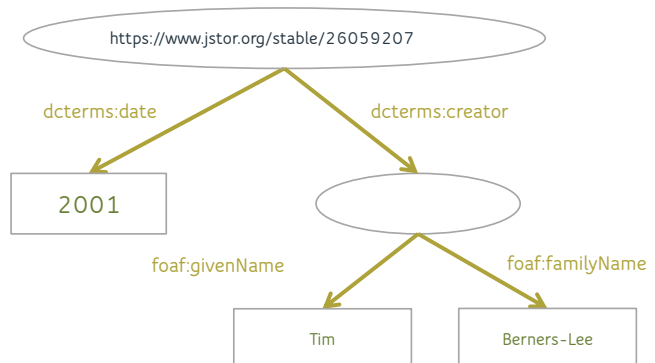
```
{
  ...
}
```


Syntaxe : Exemple N-Quads

```
<https://www.jstor.org/stable/26059207> <http://purl.org/dc/terms/date> "2001".
<https://www.jstor.org/stable/26059207> <http://purl.org/dc/terms/creator>
<https://www.w3.org/People/Berners-Lee/> <http://www.example.org/Articles>.
... <http://www.example.org/Livres>.
```

Exercice

- Écrire le graphe RDF suivant dans les syntaxes RDF/XML et Turtle



Syntaxe : Solution

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms#"
  xmlns:foaf="http://xmlns.com/foaf/spec">

  <rdf:Description rdf:about="https://www.jstor.org/stable/26059207"
    dcterms:date="2001">
    <dcterms:creator>
      <rdf:Description foaf:givenName="Tim"
        foaf:familyName="Berners-Lee"/>
    </dcterms:creator>
  </rdf:Description>
</rdf:RDF>
```

LINKED OPEN DATA

99

99

Solution

- Écrire le graphe RDF suivant dans les syntaxes RDF/XML et Turtle

@prefix dcterms: <http://purl.org/dc/terms#> .

@prefix foaf: <http://xmlns.com/foaf/spec> .

<https://www.jstor.org/stable/26059207> dcterms:creator [

foaf:familyName "Berners-Lee" ;

foaf:givenName "Tim"] ; dcterms:date "2001" .

Remarque : vous pouvez utiliser ce lien pour passer d'un format à un autre : <https://www.easyrdf.org/converter>

LINKED OPEN DATA

100

100

RDF : limites de représentation

- RDF permet de décrire les **ressources** et leurs **relations** de manière **formelle** et **structurée** sous forme de **triplets** RDF.
 - Par exemple, une **ressource** peut avoir plusieurs **propriétés** et **relations**, et il peut être difficile de les organiser et de les comprendre simplement à l'aide de triplets RDF
 - Cette présentation ne suffit pour exprimer la **signification complète des données**, car une ressource peut avoir **plusieurs propriétés et relations**.
- C'est là qu'intervient le **besoin d'une représentation sémantique**.

LINKED OPEN DATA

101

101

Représentation sémantique

- Une **représentation sémantique** permet d'organiser les données de manière plus complexe et de capturer la signification complète des ressources et de leurs relations:

- Utilisée pour décrire les classes, les propriétés et les relations entre les ressources.
- Permet une compréhension plus profonde et plus structurée des données.

Ontologie

- classification hiérarchique des entités qui permet de les organiser en fonction de leur parenté et de leur niveau de généralité.
- Par exemple, la taxonomie du règne animal organise les espèces en fonction de leur parenté, de la plus générale à la plus spécifique.

Taxonomie

LINKED OPEN DATA

102

102



Ontologie

LINKED OPEN DATA

103

103

Ontologie

- L'ontologie pour la connaissance est l'équivalent du dictionnaire pour le langage:
 - Elle contient l'ensemble **des catégories conceptuelles** utilisées pour exprimer nos connaissances sur le monde ou sur un domaine.
 - Elle contient les **définitions des catégories**, les **contraintes sémantiques/conceptuelles** sur les catégories, et éventuellement les utilisations typiques des catégories.
 - Généralement, elle organise toutes les catégories en une **hiérarchie de généralisation/spécialisation**.

A. Kabbaj Note de cours Intelligence Artificielle INSEA, 2016/2017

LINKED OPEN DATA

104

104

Ontologie

- En Informatique et sciences de l'information:
 - « Une ontologie définit **les termes** et **les relations** de base du vocabulaire d'un domaine ainsi que **les règles** qui indiquent comment combiner les termes ⁽¹⁾ et les relations de façon à pouvoir **étendre le vocabulaire** »
 - « *spécification explicite d'une conceptualisation* » ⁽²⁾ dans le but de la rendre **utilisable par la machine**

1. Neches, Robert, et al. "Enabling technology for knowledge sharing." *AI magazine* 12.3 (1991): 36-36.

2.T.R. Gruber, "A translation approach to portable ontology specifications." *Knowledge acquisition* 5.2 (1993): 199-220.

Ontologie

- «Une Ontologie définie :
 - un ensemble de **primitives représentatives** avec lesquels on modélise un domaine de connaissance ou de discours.
 - Les primitives représentatives sont typiquement :
 - des classes,
 - des attributs
 - et des relations».

Tom Gruber

Ontologie : Concept

- Représentation générale et abstraite d'un objet ou d'un ensemble d'objets ayant des caractères communs (les attributs)
- Construit à partir d'un ensemble de constructeurs qui définissent les rôles associés au concept et les restrictions qui leur sont attachées
- Peut être atomique (primitif) ou défini (composé)
 - **Exemples** : Humain, Homme, Femme, Père, Mère, Meuble, Planète, Livre, Grand-père, Grand-mère, Famille, ...
- Un concept exprimé dans une langue est un terme
- Un même concept peut avoir plusieurs termes (Humain = , **إنسان**Human, ...)

LINKED OPEN DATA

107

107

Ontologie: Relation

- Les concepts peuvent être liés entre eux par **des relations autonomes**
 - **Hiérarchiques** : lie un concept supérieur à un concept inférieur; ce dernier a les mêmes propriétés avec au moins une de plus (ex: Humain,Père)
 - **Sémantiques** : lie un concept à un autre concept avec la relation "Partie de" (ex: Page, Livre)

LINKED OPEN DATA

108

108

Avantages des ontologies

- Réutilisation du savoir sur un domaine donné
- Clarification et description explicite des concepts et des relations entre eux, bénéfique notamment pour les nouveaux utilisateurs.
- Distinction entre le savoir et le savoir opérationnel dans un domaine donné
- Facilitation de la communication humain à humain, humain à machine et machine à machine...

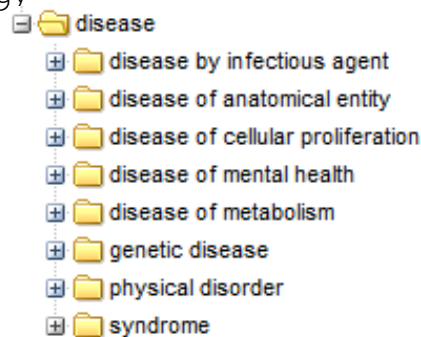
LINKED OPEN DATA

109

109

Exemples d'ontologies

- The Disease Ontology



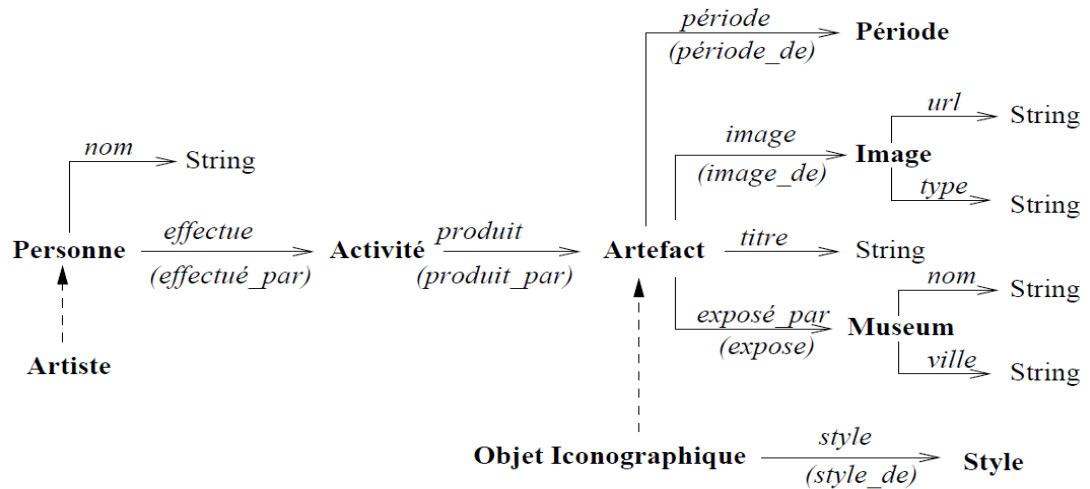
LINKED OPEN DATA

<https://disease-ontology.org/>

110

110

Exemples d'ontologies

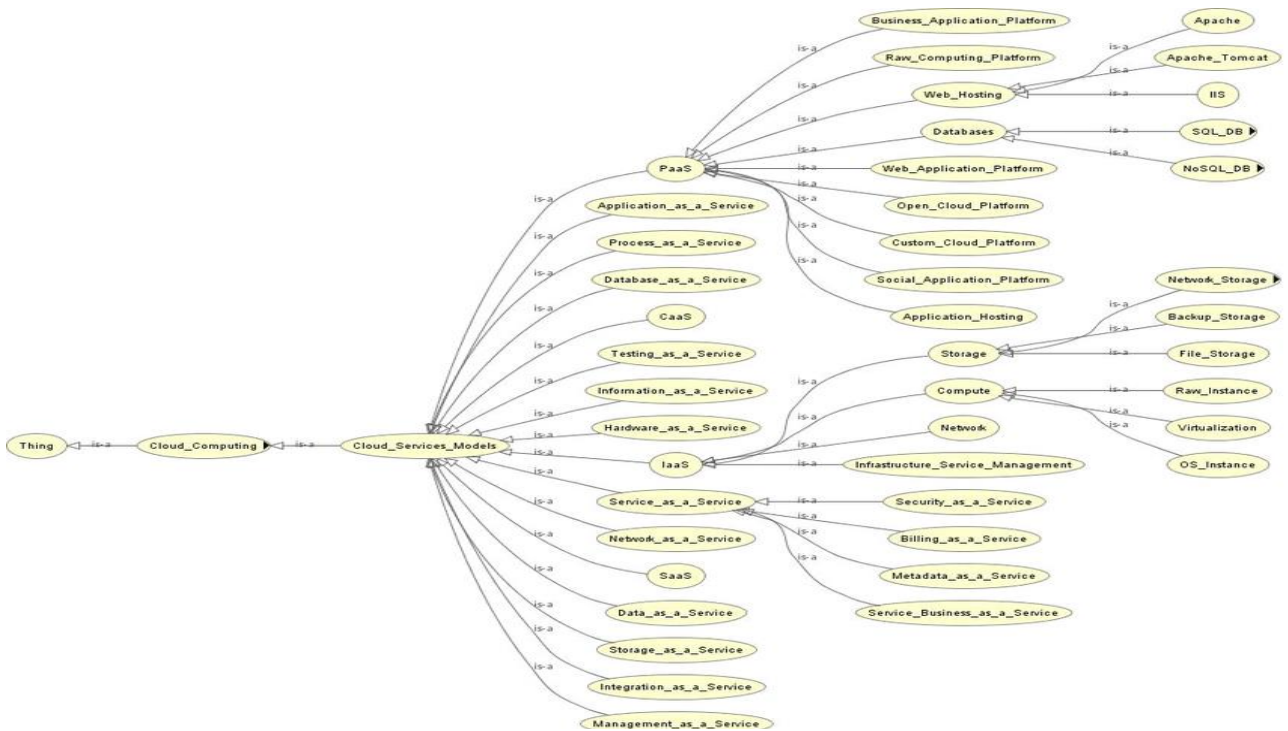


LINKED OPEN DATA

B. Amann, « Cours No 4 - Le Web Sémantique », INRIA

111

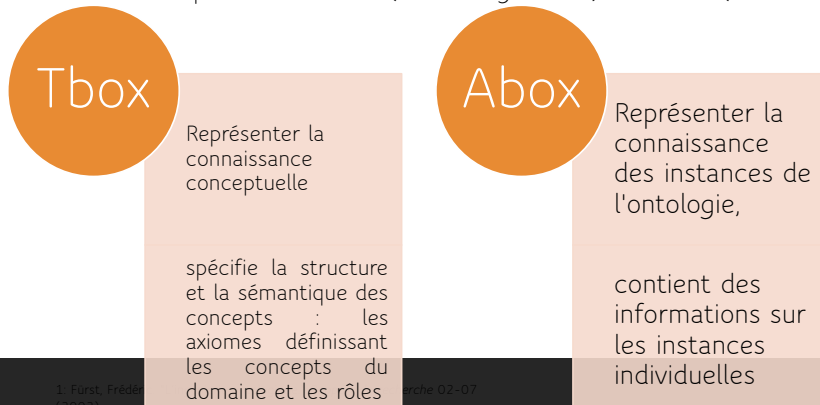
111



112

Ontologie: Base de connaissances

- L'ontologie est une base de connaissances qui permet de représenter de manière précise et claire les concepts et leurs liens avec le monde réel.
- Elle contient deux composantes : la **TBox** (Terminological Box) et la **Abox** (Assertion Box)



113

Ontologie: Base de connaissances / raisonnement

- Le raisonnement dans l'ontologie permet de :
 - vérifier la **cohérence des assertions avec les connaissances existantes**.
 - l'inférence de faits implicites à partir des connaissances existantes.
 - générer de nouvelles connaissances.
 - facilite la communication et la réutilisation dans différents domaines.
- Les raisonneurs tels que ceux utilisés dans Protégé peuvent utiliser la Tbox et l'Abox pour effectuer des raisonnements et des inférences sur les connaissances représentées dans l'ontologie.

114

Classification des ontologies

- Les ontologies peuvent être classifiées selon
 - Le formalisme utilisé pour les modéliser
 - L'objet de conceptualisation

LINKED OPEN DATA

115

115

Classification selon formalisme

Ontologies informelles

- Exprimées en langage naturel

Ontologies semi-informelles

- Écrites en langage naturel mais en respectant une certaine structuration

Ontologies semi-formelles

- Écrites dans un langage défini de manière formelle

Ontologies formelles

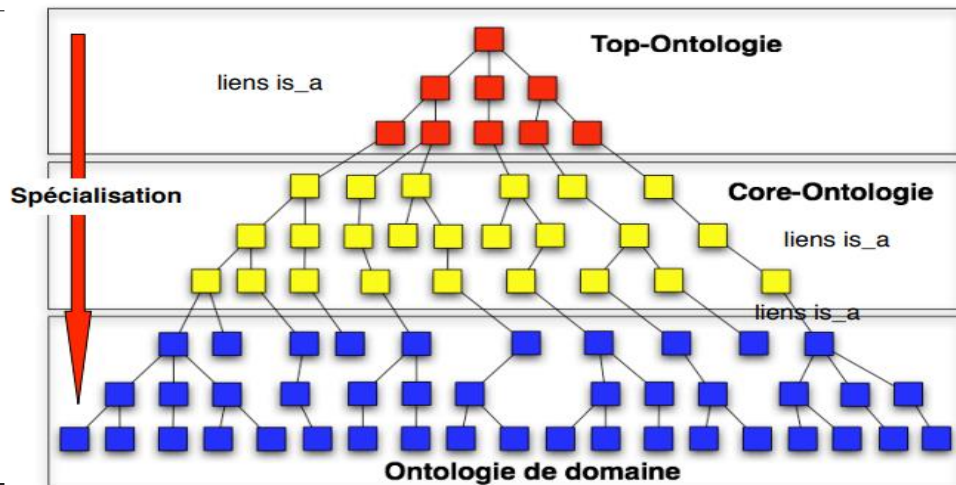
- Écrites dans un langage défini de manière formelle avec des théorèmes et preuves sur leurs propriétés

LINKED OPEN DATA

116

116

Classification par objet de conceptualisation



LINKED OPEN DATA

117

117

Classification par objet de conceptualisation

- Décrit des concepts très généraux et abstraits communs à tous les domaines

Ontologies
supérieures (top-
level ontologies)

- Décrit des concepts généraux (moins abstraits que top-level) communs à plusieurs domaines
- Exemple : WordNet

Ontologies
génériques (méta-
ontologies)

LINKED OPEN DATA

118

118

Classification par objet

- Décrivent de manière détaillée les tâches d'une activité donnée ou pour la résolution d'un problème donné indépendamment du domaine

Ontologies de tâches (task ontologies)

- Modélisent les concepts et relations pour un domaine précis (la plupart des ontologies existantes)

Ontologies de domaine (domain ontologies)

Classification par objet

- Décrivent de manière détaillée les tâches d'une activité donnée ou pour la résolution d'un problème donné dans un domaine précis

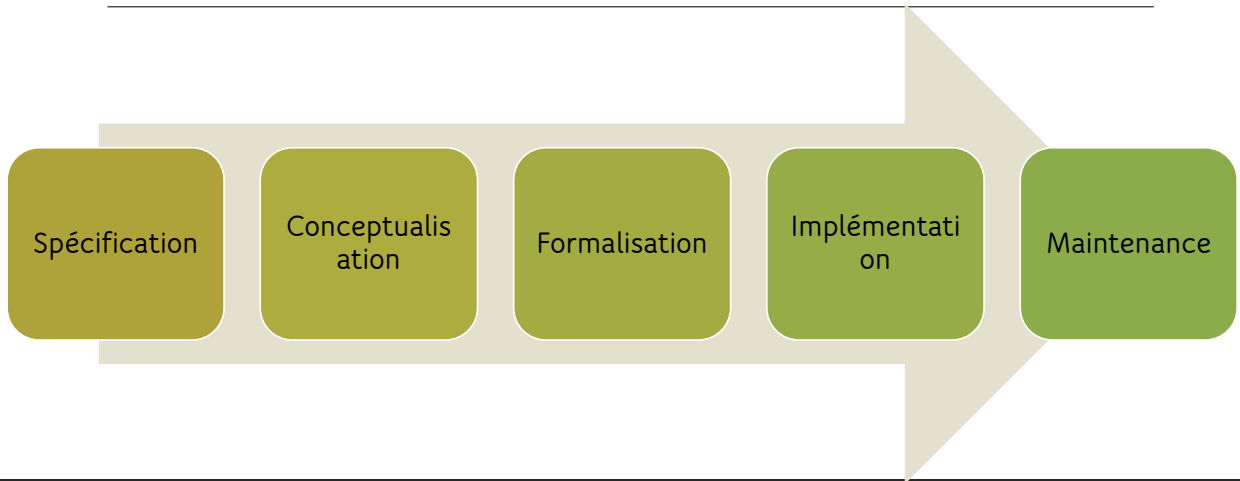
Ontologies de tâches-domaines

- Décrivent les concepts et relations d'un domaine particulier pour une application données. Ce sont les ontologies les plus spécifiques

Ontologies d'application (application ontologies)

Ontologies

principales phases de construction par la méthode : **METHONTOLOGY**



LINKED OPEN DATA

121

121

Ontologies

principales phases de construction par la méthode : **METHONTOLOGY**

- **Spécification** : Consiste à préciser l'objet global de l'ontologie, en répondant à ces questions :
 - Quel est le domaine que va couvrir l'ontologie ?
 - Dans quel but utiliserons-nous l'ontologie ?
 - A quels types de questions l'ontologie devra-t-elle fournir des réponses ?
 - Qui va utiliser et maintenir l'ontologie ?
- **Conceptualisation** : Cette phase de conceptualisation représente la pierre angulaire du développement des ontologies.
 - elle revient à structurer et à organiser les connaissances pour aboutir à un modèle conceptuel formel.
 - Parmi les livrables de cette étape sont : le dictionnaire des concepts, les taxonomies de concepts, diagrammes, glossaires des termes ;

LINKED OPEN DATA

122

122

Ontologies

principales phases de construction par la méthode METHONTOLOGY

- Formalisation :
 - Permet de formaliser l'ontologie conceptuelle obtenue à l'étape précédente afin de faciliter sa représentation ultérieure dans un langage formel ;
- Implémentation :
 - Consiste à construire un modèle opérationnel en utilisant un langage formel ;
- Maintenance :
 - mise à jour de l'ontologie.

Ontologie : Outils

- OntoEdit
- KAON
- PROTEGE (<http://protege.stanford.edu/>)

Ontologie : Langages de représentation

- RDFS: Resource Description Framework Schema
- OWL: Ontology Web Language

LINKED OPEN DATA

125

125

Exemples d'ontologie

- Suggested Upper Merged Ontology (SUMO) : <https://www.ontologyportal.org/>

LINKED OPEN DATA

126

126



Logiciel protégé / RDFS & OWL

LINKED OPEN DATA

127

127

RDFS : Resource Description Framework Schema

- Langage de documentation du vocabulaire utilisé dans les données RDF: **méta-vocabulaire**
- RDFS est le langage de description des vocabulaires RDF
- Un vocabulaire RDF permet de définir les classes et les propriétés d'un schéma RDF
- Permet de nommer les **classes** et **propriétés** et d'en définir une organisation hiérarchique
- Modélisé en RDF
- Interrogé avec SPARQL

LINKED OPEN DATA

128

128

RDFS : Resource Description Framework Schema

Vocabulaire léger permettant de décrire de petites ontologies légères afin d'assurer l'interopérabilité entre les systèmes qui les échangent

Recommandation du W3C

◦ RDFS 1.0 (2004) RDFS 1.1 (2014)

LINKED OPEN DATA

129

129

RDFS

- Un vocabulaire RDF est décrit en RDF :
 - la propriété **rdfs:subClassOf** permet de définir **des héritages**
 - **Un URI peut représenter une classe** s'il est associé à la propriété **rdf:type** qui a la valeur **rdfs:Class**.
 - **Un URI représente une propriété** si la propriété **rdf:type** a la valeur **rdf:Property**, qui est la métaclasse des propriétés.
 - On peut utiliser les propriétés d'annotation : **rdfs:label**, **rdfs:comment**, **rdfs:isDefinedBy**, **rdfs:seeAlso**

LINKED OPEN DATA

130

130

Classes RDFS

| Class name | comment |
|--|---|
| rdfs:Resource | The class resource, everything. |
| rdfs:Literal | The class of literal values, e.g. textual strings and integers. |
| rdf:langString | The class of language-tagged string literal values. |
| rdf:HTML | The class of HTML literal values. |
| rdf:XMLLiteral | The class of XML literal values. |
| rdfs:Class | The class of classes. |
| rdf:Property | The class of RDF properties. |
| rdfs:Datatype | The class of RDF datatypes. |
| rdf:Statement | The class of RDF statements. |
| rdf:Bag | The class of unordered containers. |
| rdf:Seq | The class of ordered containers. |
| rdf:Alt | The class of containers of alternatives. |
| rdfs:Container | The class of RDF containers. |
| rdfs:ContainerMembershipProperty | The class of container membership properties, <code>rdfs:_1</code> , <code>rdfs:_2</code> , ..., all of which are sub-properties of 'member'. |
| rdf:List | The class of RDF Lists. |

https://www.w3.org/TR/rdf-schema/#ch_4umclasses

LINKED OPEN DATA

131

131

Classes RDFS rdfs:Property

- C'est la classe de toutes les propriétés
- C'est une instance de `rdfs:Class`
- Lorsque l'on instancie cette `rdfs:Property`, on peut renseigner :
 - `rdfs:domain` : la source de la propriété
 - `rdfs:range` : la valeur de cette propriété (Si non renseigné, alors tout est accepté)

```
<rdfs:Class rdfs:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property">
  <rdfs:isDefinedBy rdfs:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
  <rdfs:label>Property</rdfs:label>
  <rdfs:comment>The class of RDF properties.</rdfs:comment>
  <rdfs:subClassOf rdfs:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>
```

Exemples de création de propriétés

```
<rdf:Property rdfs:about="http://www.w3.org/2000/01/rdf-schema#subClassOf">
  <rdfs:isDefinedBy rdfs:resource="http://www.w3.org/2000/01/rdf-schema#"/>
  <rdfs:label>subClassOf</rdfs:label>
  <rdfs:comment>The subject is a subclass of a class.</rdfs:comment>
  <rdfs:range rdfs:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:domain rdfs:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Property>

<rdf:Property rdfs:about="http://www.w3.org/2000/01/rdf-schema#comment">
  <rdfs:isDefinedBy rdfs:resource="http://www.w3.org/2000/01/rdf-schema#"/>
  <rdfs:label>comment</rdfs:label>
  <rdfs:comment>A description of the subject resource.</rdfs:comment>
  <rdfs:domain rdfs:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:range rdfs:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
```

LINKED OPEN DATA

132

132

Quelques propriétés

| Quelques propriétés | | | |
|---------------------|---------------------------------|----------------------------|----------------------------|
| | Property name | domain | range |
| | <code>rdf:type</code> | <code>rdfs:Resource</code> | <code>rdfs:Class</code> |
| | <code>rdfs:subClassOf</code> | <code>rdfs:Class</code> | <code>rdfs:Class</code> |
| | <code>rdfs:subPropertyOf</code> | <code>rdf:Property</code> | <code>rdf:Property</code> |
| | <code>rdfs:domain</code> | <code>rdf:Property</code> | <code>rdfs:Class</code> |
| | <code>rdfs:range</code> | <code>rdf:Property</code> | <code>rdfs:Class</code> |
| | <code>rdfs:label</code> | <code>rdfs:Resource</code> | <code>rdfs:Literal</code> |
| | <code>rdfs:comment</code> | <code>rdfs:Resource</code> | <code>rdfs:Literal</code> |
| | <code>rdfs:seeAlso</code> | <code>rdfs:Resource</code> | <code>rdfs:Resource</code> |
| | <code>rdfs:isDefinedBy</code> | <code>rdfs:Resource</code> | <code>rdfs:Resource</code> |
| | <code>rdf:value</code> | <code>rdfs:Resource</code> | <code>rdfs:Resource</code> |
| | <code>rdf:subject</code> | <code>rdf:Statement</code> | <code>rdfs:Resource</code> |

LINKED OPEN DATA

133

133

Propriétés RDFS

`rdfs:domain` / `rdfs:range`

- Toute ressource qui a une propriété `rdfs:range` est une instance de `rdf:Property` (i.e. `rdfs:domain` de `rdfs:range` est `rdf:Property`)
- Toute ressource qui est une valeur d'une propriété `rdfs:domain` est une instance de `rdfs:Class` (i.e. `rdfs:range` de `rdfs:domain` est `rdfs:Class`)

LINKED OPEN DATA

134

134

Propriétés RDFS

rdfs:range

- rdfs:range permet de définir le type de valeurs que cette propriété peut prendre

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix ex: <http://example.com/> .

ex:Auteur a rdfs:Class .

ex:aEteEcritPar a rdf:Property ;
    rdfs:domain ex:Livre ;
    rdfs:range ex:Auteur .
```

toutes les valeurs possibles de cette propriété doivent être des instances de la classe "Auteur"

LINKED OPEN DATA

135

135

Propriétés RDFS

rdfs:domain

- "rdfs:domain" est utilisée pour spécifier le domaine d'une propriété dans un vocabulaire RDF. Elle est utilisée pour indiquer la classe ou le type de données du sujet auquel une propriété peut être appliquée.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix ex: <http://example.com/> .

ex:Auteur a rdfs:Class .

ex:aEteEcritPar a rdf:Property ;
    rdfs:domain ex:Livre ;
    rdfs:range ex:Auteur .
```

Cela indique que la propriété « aEteEcritPar » ne peut être appliquée qu'aux instances de la classe « livre », qui représente les livres dans votre vocabulaire RDF.

LINKED OPEN DATA

136

136

Propriétés RDFS

rdfs:domain / rdfs:range



```

<rdf:Property rdf:ID="estPère">
  <rdfs:range rdf:resource="#Personne"/>
  <rdfs:domain rdf:resource="#Homme"/>
</rdf:Property>
  
```

LINKED OPEN DATA

137

137

Propriétés RDFS

rdf:type

- Spécifie qu'une ressource est une instance d'une classe
- Est instance de *rdf:Property*

■ *R* *rdf:type* *C*

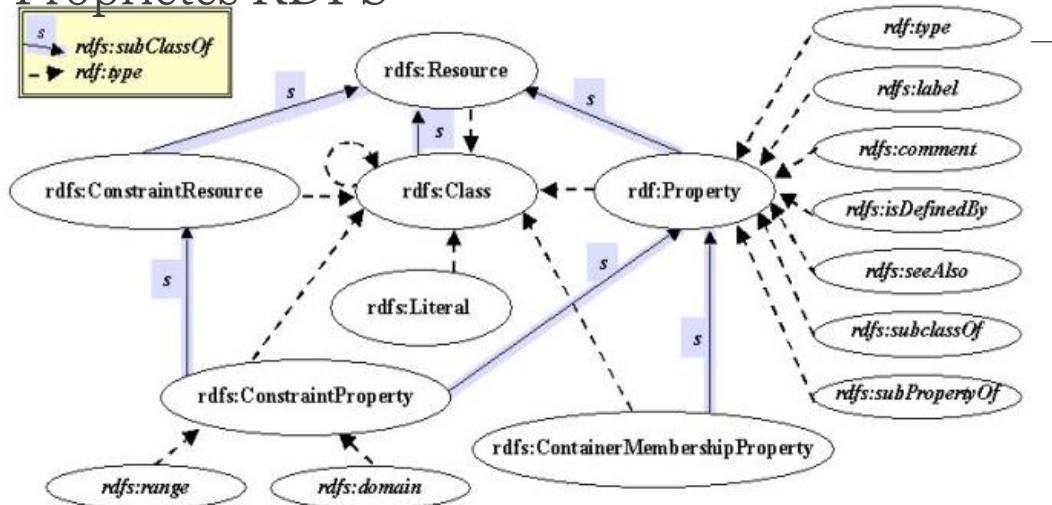
- *R* est une instance de la classe *C*
- *C* est une instance de la classe *rdfs:Class*

LINKED OPEN DATA

138

138

Propriétés RDFS



<https://www.w3.org/2001/sw/RDFCore/Schema/200109>

LINKED OPEN DATA

139

139

Hiérarchies dans RDFS

Hiérarchies des classes : propriétés

- Toute classe est sous-classe de $rdfs:Resource$
 - $C \text{ rdf:type } rdfs:Class \Rightarrow C \text{ rdfs:subClassOf } rdfs:Resource$
- Les types sont propagés
 - $C_1 \text{ rdfs:subClassOf } C_2 \text{ et } S \text{ rdf:type } C_2 \Rightarrow S \text{ rdf:type } C_1$
- La subsomption est transitive
 - $C_1 \text{ rdfs:subClassOf } C_2 \text{ et } C_2 \text{ rdfs:subClassOf } C_3 : \Rightarrow C_1 \text{ rdfs:subClassOf } C_3$
- La subsomption est réflexive
 - $C \text{ rdf:type } rdfs:Class \Rightarrow C \text{ rdfs:subClassOf } C$

LINKED OPEN DATA

140

140

Documentation des schémas

- Utilisation de labels
- Utilisation de commentaires
- Liaison entre des classes (resp. des propriétés) en utilisant *rdfs:seeAlso*

```
<rdf:RDF xml:base="http://example.org/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <rdfs:Class rdf:ID="Tennis">
    <rdfs:subClassOf rdf:resource="#Individuel"/>
    <rdfs:subClassOf rdf:resource="#Sport"/>
    <rdfs:seeAlso rdf:resource="#RolandGarros"/>
  </rdfs:Class>
</rdf:RDF>
```

LINKED OPEN DATA

141

141

Limitations de RDF/RDF-S

- **rdfs:range** définit le domaine de valeurs d'une propriété quelle que soit la classe concernée :
 - Ex : il ne permet pas d'exprimer que certains animaux ne mangent que de l'herbe alors que d'autres sortes mangent également de la viande.
- **RDF-S** ne permet pas d'exprimer que 2 classes sont disjointes :
 - Ex : les classes des hommes et des femmes sont disjointes.
- **RDF-S** ne permet pas de créer des classes par combinaison ensembliste d'autres classes (intersection, union, complément) :
 - Ex : on veut construire la classe Personne comme l'union disjointe des classes des hommes et des femmes.

LINKED OPEN DATA

142

142

Limitations de RDF/RDF-S

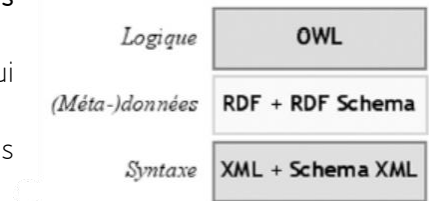
- RDF-S ne permet pas de définir de restriction sur le nombre d'occurrences de valeurs que peut prendre une propriété :
 - Ex : on ne peut pas dire qu'une personne a exactement 2 parents.
- RDF-S ne permet pas de caractériser des propriétés notamment :
 - transitivité : Ex : estPlusGrandQue
 - unicité : Ex : estLePèreDe
 - propriété inverse : Ex : « mange » = propriété inverse de « estMangéPar »

Limitations de RDF/RDF-S

- Il reste un langage limité qui ne permet pas de représenter des ontologies grandes/complexes
 - Définir des classes par restrictions sur des propriétés
 - Définir des classes disjointes
 - Définir des classes par conditions booléennes
 - Caractérisation de propriétés
- RDFS reste un langage limité qui ne permet pas de représenter des ontologies grandes/complexes

OWL: Ontology Web Language

- OWL est un langage d'ontologies pour le Web Sémantique qui permet de représenter des ontologies riches et complexes. Il permet de :
 - faciliter le raisonnement d'agents logiciels qui accèdent à des ressources sur le Web
 - permettre l'utilisation de plusieurs ontologies pouvant être reliées
- OWL Recommandation du W3C:
 - OWL 1 (2004), OWL 2 (2012)



LINKED OPEN DATA

145

145

Description des ontologies

Une ontologie est une ressource, elle est donc identifiée par un URI

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
  <owl:Ontology rdf:about="http://example.org/myOntology/">
    ...
  </owl:Ontology>
</rdf:RDF>
```

- Communément appelé "ontology header" et placé vers le début du document de description

LINKED OPEN DATA

146

146

Description des ontologies

OWL offre des propriétés pour décrire les ontologies

- *owl:imports*, *owl:versionInfo*, *owl:priorVersion*,
owl:backwardCompatibleWith, *owl:incompatibleWith*
- En plus des propriétés d'annotation de RDFS *rdfs:label*, *rdfs:comment*,
rdfs:seeAlso, *rdfs:isDefinedBy*

LINKED OPEN DATA

147

147

OWL

Peut être vu comme une extension du RDFS, avec un pouvoir d'expression plus grand

Il permet de décrire des classes à partir :

- d'autres classes
- d'énumération d'individus
- de restrictions sur les objets des propriétés
- de cardinalités sur des propriétés

LINKED OPEN DATA

148

148

OWL

Il permet de **décrire des propriétés** à partir :

- d'autres propriétés
- en leur attribuant des qualités

Il permet de **décrire des assertions** concernant des classes, des propriétés, des individus

Création d'une nouvelle classe

- Comme en RDF, une ressource/individu est instance d'une classe avec la possibilité de mettre les individus en relation Réutilisation des **propriétés** `rdfs:subClassOf`, `rdfs:Property`, `rdfs:subProperty`, `rdfs:range` et `rdfs:domain`
- Création d'une nouvelle classe à l'aide de `owl:Class`
- Possibilité de définir une classe comme étant :
 - l'union de deux classes : `owl:unionOf`
 - l'union disjointe de plusieurs classes : `owl:disjointUnionOf` (exemple Enfant est l'union disjointe de Garçon et Fille)
 - l'intersection de deux classes : `owl:intersectionOf`
 - l'énumération des instances d'une classe : `owl:oneOf`
 - le complément d'une classe : `owl:complementOf`

OWL

Espace de nom <http://www.w3.org/2002/07/owl#>

Préfixe owl

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:example="http://example.org/">
  ...
</rdf:RDF>
```

LINKED OPEN DATA

151

151

Classes owl: Axiomes de classes

Les descriptions de classes constituent la base de la définition des classes à travers des **axiomes** (*class axioms*)

La forme la plus simple d'un axiome de classe

```
<owl:Class rdf:ID="ID_Classe"/>
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:example="http://example.org/">

  <owl:Class rdf:ID="Personne"/>

</rdf:RDF>
```

LINKED OPEN DATA

152

152

Axiomes de classes

Trois principaux concepts

- *rdfs:subClassOf* (même que

RDFS)

- *owl:equivalentClass*

- *owl:disjointWith*

```
<owl:Class rdf:ID="Humain">
  <owl:equivalentClass rdf:resource="foaf:Person"/>
</owl:Class>

<owl:Class rdf:ID="1A_GroupeA">
  <owl:disjointWith rdf:resource= #1A_GroupeB/>
</owl:Class>
```

LINKED OPEN DATA

153

153

Classes OWL

Déclarées par *owl:Class*

Une classe peut être décrite par

- Énumération exhaustive
- Union de deux classes ou plus
- Intersection de deux classes ou plus
- Négation
- Restriction de propriétés

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:example="http://example.org/">
  <owl:Class rdf:ID="Continent">
    <owl:oneOf rdf:parseType="Collection">
      <owl:Thing rdf:about="#Africa"/>
      <owl:Thing rdf:about="#Asia"/>
      <owl:Thing rdf:about="#Europe"/>
      <owl:Thing rdf:about="#Australia"/>
      <owl:Thing rdf:about="#America"/>
      <owl:Thing rdf:about="#Antarctica"/>
    </owl:oneOf>
  </owl:Class>
</rdf:RDF>
```

LINKED OPEN DATA

154

154

Description de classes

Union de deux classes ou plus

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:example="http://example.org/">

  <owl:Class rdf:ID="Etudiants">
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#1ACI"/>
      <owl:Class rdf:about="#2ACI"/>
      <owl:Class rdf:about="#3ACI"/>
    </owl:unionOf>
  </owl:Class>

</rdf:RDF>
```

LINKED OPEN DATA

155

155

Description de classes

Intersection de deux classes ou plus

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:example="http://example.org/">

  <owl:Class rdf:ID="Etudiantes_2A_ICSD">
    <owl:intersectionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#2ACI"/>
      <owl:Class rdf:about="#ICSD"/>
      <owl:Class rdf:about="#Femme"/>
    </owl:intersectionOf>
  </owl:Class>

</rdf:RDF>
```

LINKED OPEN DATA

156

156

Description de classes

Négation

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:example="http://example.org/">

  <owl:Class rdf:ID="NonEtudiants">
    <owl:complementOf rdf:resource="#Etudiants"/>
  </owl:Class>

</rdf:RDF>
```

LINKED OPEN DATA

157

157

Propriétés

Deux principales catégories

- *owl:ObjectProperty* : permet de lier des ressources entre elles
- *owl:DatatypeProperty* : permet de lier des ressources à des valeurs littérales

LINKED OPEN DATA

158

158

Axiomes de propriétés

La forme la plus simple d'un axiome de propriété

```
<owl:ObjectProperty rdf:ID="ID_Prop"/>
```

LINKED OPEN DATA

159

159

Axiomes de propriétés

Principaux concepts

- Concepts RDFS : *rdfs:subPropertyOf*, *rdfs:domain*, *rdfs:range*
- Relations avec d'autres propriétés
- Contraintes sur la cardinalité
- Contraintes logiques

LINKED OPEN DATA

160

160

Axiomes de propriétés

- Relations avec d'autres propriétés
 - *owl:equivalentProperty* : indique que deux propriétés expriment exactement la même relation

LINKED OPEN DATA

161

161

Restriction de propriétés

Permet de décrire des classes anonymes dont les individus satisfont une restriction particulière

Cette restriction peut être sur les **valeurs** ou sur la **cardinalité**

LINKED OPEN DATA

162

162

Caractérisation des classes

- Il est possible d'indiquer que des classes seront toujours disjointes : `owl:AllDisjointesClasses`
- Il est possible d'indiquer que deux classes sont équivalentes : `owl:equivalentClass`

Création d'une propriété

- Propriété dont le range est une ressource : `owl:ObjectProperty`
- Propriété dont le range est un type simple : `owl:DatatypeProperty`
- Propriété permettant de décrire la ressource `owl:AnnotationProperty`

Caractérisation des propriétés

```

owl:equivalentProperty :  $P(x, y) \Rightarrow Q(x, y)$ 
owl:inverseOf :  $P(x, y) \Rightarrow P'(y, x)$ 
owl:TransitiveProperty :  $P(x, y) \wedge P(y, z) \Rightarrow P(x, z)$ 
owl:SymmetricProperty (respect. owl:AsymmetricProperty) :  $P(x, y) \Rightarrow P(y, x)$ 
(respect.  $P(x, y) \Rightarrow \neg P(y, x)$ )
owl:propertyDisjointWith :  $P(x, y) \Rightarrow \neg P'(x, y)$ 
owl:ReflexiveProperty (respect. owl:IrreflexiveProperty) :  $P(x, y) \Rightarrow x = y$  (respect.
 $P(x, y) \Rightarrow x \neq y$ )
owl:FunctionalProperty :  $P(x, y) \wedge P(x, z) \Rightarrow y = z$ 
owl:InverseFunctionalProperty :  $P(x, z) \wedge P(y, z) \Rightarrow x = y$ 
owl:hasKey (les clés sont des littéraux) :  $P(x, k_1) \wedge P(y, k_2) \wedge k_1 = k_2 \Rightarrow x = y$ 
owl:propertyChainAxiom :  $P_1(x_1, x_2) \wedge P_2(x_2, x_3) \wedge \dots \wedge P_n(x_n, x_{n+1}) \Rightarrow Q(x_1, x_{n+1})$ 

```

LINKED OPEN DATA

165

165

Caractérisation d'individus

- owl:differentFrom : deux individus sont obligatoirement différents
- owl:sameAs : deux individus sont identiques
- owl:NegativePropertyAssertion : deux individus ne sont pas reliés par une propriété P (monde ouvert)

LINKED OPEN DATA

166

166

Restriction

- `owl:Restriction` permet de créer des classes anonymes
- On l'utilise avec `rdfs:subClassOf` ou `owl:equivalentClass`

LINKED OPEN DATA

167

167

Restriction des cibles d'une propriété

- `owl:allValuesFrom` permet de définir une classe telle qu'un individu x appartient à cette classe ssi tous les individus y reliés à x (en tant qu'objet) par une propriété P ($P(x, y)$), sont tous des individus d'une classe C : $\forall y, P(x, y) \Rightarrow y \in C$
- `owl:someValuesFrom` : permet de définir une classe telle qu'un individu x appartient à cette classe ssi il existe au moins un individu y reliés à x (en tant qu'objet) par une propriété P , qui est un individu de la classe C : $\exists y \in C, P(x, y)$
- `owl:hasValue` : permet de définir une classe telle qu'un individu x appartient à cette classe ssi x est relié à la valeur v par la propriété P

LINKED OPEN DATA

168

168

Restriction de la cardinalité d'une propriété

- **owl:minCardinality** permet de définir une classe telle qu'un individu x appartient à cette classe ssi il y a au moins n propriétés P issues de x
- **owl:maxCardinality** permet de définir une classe telle qu'un individu x appartient à cette classe ssi il y a au plus n propriétés P issues de x
- **owl:cardinality** permet de définir une classe telle qu'un individu x appartient à cette classe ssi il y a n propriétés P issues de x
- **owl:minQualifiedCardinality**, **owl:maxQualifiedCardinality**, **owl:qualifiedCardinality** permet de définir une classe telle qu'un individu x appartient à cette classe ssi le nombre n de propriétés P issues de x et reliant x à des individus d'une classe C est contraint (min, max ou égale)

LINKED OPEN DATA

169

169

Exercice

En considérant l'ontologie ci-après

1. Représenter les classes sous forme de schéma
2. Compléter l'ontologie pour représenter les classes pères et enfants
3. Représenter les données suivantes en utilisant l'ontologie décrite

Sara est la mère de Amal et Karim.

Ali est le père de Sara.

LINKED OPEN DATA

170

170

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix ex: <http://www.exemple.com/>.
```

```
ex:Woman rdfs:subClassOf ex:Person .
```

```
ex:Mother
  owl:equivalentClass [
    rdf:type    owl:Class ;
    owl:intersectionOf ( ex:Woman ex:Parent )
  ] .
```

```
ex:Parent
  owl:equivalentClass [
    rdf:type    owl:Restriction ;
    owl:onProperty    ex:hasChild ;
    owl:allValuesFrom    ex:Person
  ] .
```

171

171

Comment récupérer des connaissances à partir d'une ontologie et de données RDF sur le web sémantique ?

SPARQL

LINKED OPEN DATA

172

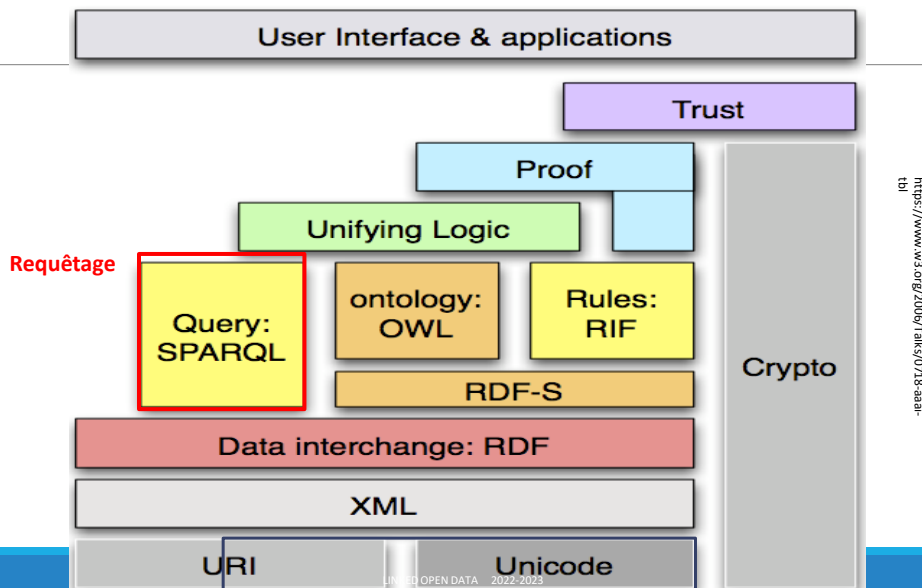
172

Comment accéder aux données RDF ?

Nécessité d'avoir un langage de requête comme XQuery pour XML et SQL pour les bases de données relationnelles

173

Qu'est-ce que SPARQL ?



174

Qu'est-ce que SPARQL ?

SPARQL Protocol And RDF Query Language

- Initialement **Simple Protocol And RDF Query Language**^(*)

Langage d'interrogation des graphes RDF composé de

- SPARQL Query Language for RDF
- SPARQL Protocol for RDF
- SPARQL Query Results XML Format

(*) <https://lists.w3.org/Archives/Public/semantic->

LINKED OPEN DATA 2022-2023

175

175

Qu'est-ce que SPARQL ?

SPARQL Query Language for RDF

- Langage de requête sur les graphes RDF qui permet de spécifier le type de données recherchées
- Recommandation du W3C
 - SPARQL 1.0 Query Language (2008)
 - SPARQL 1.1 Query Language (2013)
 - Utilisé avec SPARQL 1.1 Update qui permet d'exécuter des opérations "update" sur des graphes RDF

LINKED OPEN DATA 2022-2023

176

176

Qu'est-ce que SPARQL ?

SPARQL Query Results XML Format

- Format de représentation des résultats d'une requête SPARQL
- Recommandation du W3C
 - SPARQL Query Results XML Format (2013)

Qu'est-ce que SPARQL ?

SPARQL Protocol for RDF

- Protocole pour soumettre une requête SPARQL d'un client vers des serveurs et recevoir les résultats, notamment à travers les protocoles HTTP et SOAP
- Recommandation du W3C
 - SPARQL 1.0 Protocol for RDF (2008)
 - SPARQL 1.1 Protocol for RDF (2013)

Qu'est-ce que SPARQL ?

SPARQL permet de

- Extraire l'information sous forme de URI, de nœuds ou de littéraux
- Extraire des sous-graphes RDF
- Construire de nouveaux graphes RDF à partir de l'information extraite

Qu'est-ce que SPARQL ?

Quelques endpoints SPARQL

| Endpoint | URL | Description |
|----------|---|---|
| SPARQLer | http://sparql.org/sparql.html | Requêtes sur des datasets RDF accessibles sur le Web |
| Virtuoso | http://lod.openlinksw.com/sparql/ | Requêtes sur des datasets RDF accessibles sur le Web |
| DBPedia | http://dbpedia.org/sparql | Requêtes sur DBPedia (données RDF de Wikipedia) ainsi que sur des datasets accessibles sur le Web |
| UniProt | https://sparql.uniprot.org/sparql | Requêtes sur les données de UniProt (BD de séquences de protéines) |

Plus sur <https://www.w3.org/wiki/SparglEndpoints>

Appariement de graphes

Appariement de graphe

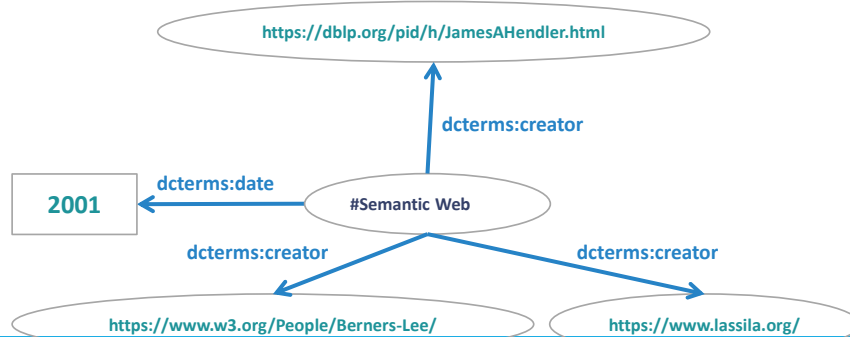
Une requête SPARQL est un graphe avec des variables

Recherche de sous-graphes dans un graphe de données
selon un motif (*pattern*) donné → **appariement de graphe**

Appariement de graphe

Exemple: qui sont les auteurs de l'article "Semantic web" ?

Graphe à interroger



LINKED OPEN DATA 2022-2023

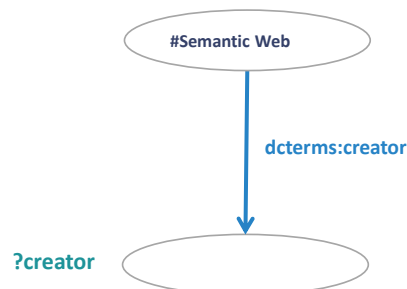
183

183

Appariement de graphe

- Exemple: qui sont les auteurs de l'article "Semantic web" ?

Graphe de la requête



LINKED OPEN DATA 2022-2023

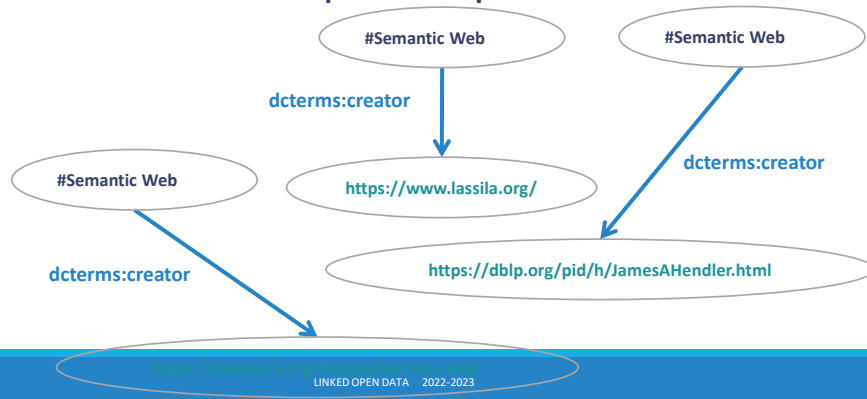
184

184

Appariement de graphe

- Exemple: qui sont les auteurs de l'article "Semantic web" ?

Graphes des réponses



185

Requête SELEct

186

Structure

Structure d'une requête SELECT

1. Déclaration des vocabulaires (**PREFIX**)
2. Identification des variables dont la valeur est retournée par la requête (**SELECT**)
3. Identification du graphe à interroger (**FROM**)
4. Précision du pattern du sous-graphe à chercher (**WHERE**)
5. Actions sur les résultats obtenus (**GROUP BY ...**)

Syntaxe

Syntaxe inspirée de la syntaxe SQL

PREFIX vocabulaires utilisés

SELECT variables à retourner (* pour toutes)

[FROM] graphe à interroger (optionnel)

WHERE contraintes à satisfaire par le graphe qui s'apparie avec le graphe à interroger

Syntaxe

Syntaxe inspirée de la syntaxe SQL

Syntaxe basée sur la syntaxe Turtle où les variables sont précédées par "?" ou "\$" et peuvent représenter tout type de nœud

Syntaxe

Le résultat d'une requête peut être stocké dans différents formats

- XML
- JSON
- CSV
- TSV
- Texte

Syntaxe

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="x"/>
    <variable name="fname"/>
    <variable name="gname"/>
  </head>
  <results>
    <result>
      <binding name="x">
        <uri>https://www.w3.org/People/Berners-Lee/card#i</uri>
      </binding>
      <binding name="fname">
        <literal>Berners-Lee</literal>
      </binding>
      <binding name="gname">
        <literal>Timothy</literal>
      </binding>
    </result>
  </results>
</sparql>
```

191

191

Filtres et fonctions

Il est possible d'ajouter des contraintes dans la clause

WHERE avec le mot-clé **FILTER**

- Logique : **!**, **&&**, **||**
- Comparaison : **<**, **<=**, **>**, **>=**, **=**, **!=**
- Opérations : **+**, **-**, *****, **/**
- Tests : *isURI(?x)*, *isBlank(?x)*, *isLiteral(?x)*, *bound(?x)*
- Accessors : *str(?x)*, *lang(?x)*, *datatype(?x)*

192

Filtres et fonctions

Il est possible d'ajouter des contraintes dans la clause

WHERE avec le mot-clé **FILTER**

- Fonctions sur les nombres
 - *abs(?x)* , *round(?x)* , *ceil(?x)* , *floor(?x)* , *rand(?x)* , *isNumeric(?x)*
- Fonctions sur les dates
 - *now()* , *year(?x)* , *month(?x)* , *day(?x)* , *hours(?x)* , *minutes(?x)* , *seconds(?x)* , *timezone(?x)* , *tz(?x)*

Filtres et fonctions

Il est possible d'ajouter des contraintes dans la clause

WHERE avec le mot-clé **FILTER**

- Fonctions sur les chaînes de caractères
 - *contains(?x , txt)* , *strstarts(?x , txt)* , *strends(?x , txt)* , *strlen(?x)* , *strlang(?x, lang)* , *langMatches(?x , lang)* , *ucase(?x)* , *lcase(?x)* , *substr(?x , start [, length])* , *encode_for_uri(?x)* , *strdt(?x, type)*
- Fonctions XPath 2.0

Filtres et fonctions

Il est possible d'ajouter des contraintes dans la clause

WHERE avec le mot-clé **FILTER**

- Test conditionnel avec **IF**
- Test de présence de valeurs avec **IN** (ou **NOT IN**)
- Test de présence d'un pattern de sous-graphe avec **EXISTS** (ou **NOT EXISTS**)

Actions sur les résultats

Il est possible d'agir sur les résultats

- **LIMIT** préciser le nombre de résultats à retourner
- **OFFSET** préciser le nombres de résultats à ignorer avant de commencer à retourner les résultats
- **ORDER BY** trier les résultats
- **GROUP BY** regrouper les résultats
- **HAVING** préciser une condition sur les résultats
- Agrégation (**count** , **sum** , **avg** , **max** , **min** ...)

Graphes nommés

Graphes nommés

Les requêtes précédentes sont exécutées sur un seul graphe ➔ **graphe par défaut**

La clause **FROM** permet de spécifier le graphe par défaut sur lequel la requête est exécutée

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?n
FROM < https://www.example.org/Person1>
WHERE { ?x foaf:givenname ?n . }
```

Graphes nommés

Les requêtes précédentes sont exécutées sur un seul graphe → **graphe par défaut**

La clause **FROM** permet de spécifier le graphe par défaut sur lequel la requête est exécutée

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?n
FROM < https://www.example.org/Person1>
FROM < https://www.example.org/Person2>
WHERE {?x foaf:givenname ?n .}
```

Graphes nommés

Les requêtes précédentes sont exécutées sur un seul graphe
→ **graphe par défaut**

La clause **FROM** permet de spécifier le graphe par défaut sur lequel la requête est exécutée

Les datasets RDF sont composés du graphe par défaut ainsi que de 0 ou n **graphes nommés** (*named graphs*) identifiés par des URIs

Graphes nommés

Les clauses **FROM NAMED** permettent de spécifier les graphes nommés

La clause **GRAPH** permet de spécifier les parties de la requête à exécuter sur les graphes nommés

Le reste de la requête est exécuté sur le graphe par défaut

LINKED OPEN DATA 2022-2023

201

201

Graphes nommés

Possibilité de préciser le graphe nommé à utiliser

PREFIX ex: <http://www.example.org/>

SELECT ?n

FROM ex:graph1

FROM NAMED ex:graph2

FROM NAMED ex:graph3

WHERE {?x ex:name ?n .

GRAPH ex:graph2 { ... }

GRAPH ex:graph3 { ... } }

LINKED OPEN DATA 2022-2023

202

202

Graphes nommés

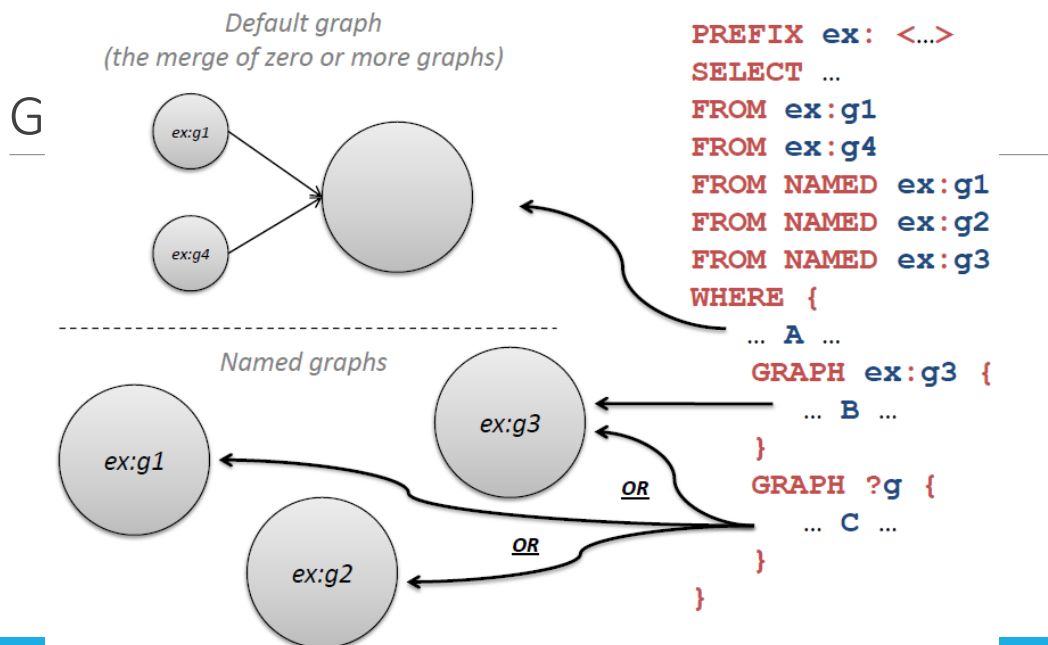
L'URI du graphe qui « match » la requête dans **GRAPH** est lié à la variable **?g**

```
PREFIX ex: <http://www.example.org/>
SELECT ?n
FROM ex:graph1
FROM NAMED ex:graph2
FROM NAMED ex:graph3
WHERE {?x ex:name ?n .
      GRAPH ?g {?x ex:name ?n .} }
```

LINKED OPEN DATA 2022-2023

203

203



Feigenbaum, Lee. "SPARQL by example: the cheat sheet." *Cambridge semantics* (2008).

A 2022-2023

204

204

Autres requêtes SPARQL

Autres requêtes SPARQL

En plus de la requête **SELECT**, SPARQL dispose de trois autres requêtes

- Requête **ASK**
- Requête **CONSTRUCT**
- Requête **DESCRIBE**

Requête ASK

Permet de vérifier s'il existe au moins un résultat pour un pattern de sous-graphe

Retourne un booléen **True/False** ou **Yes/No**

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
ASK { ?x foaf:givenname ?n . }
```

Requête CONSTRUCT

Permet de construire un nouveau graphe RDF

Retourne un graphe RDF qui est crée à partir des résultats de la requête et sous forme du pattern spécifié dans la requête

Utile pour transformer des graphes RDF (par exemple utiliser un autre vocabulaire...)

Requête DESCRIBE

Permet de décrire une ressource

Retourne un graphe RDF

Utile quand on n'a pas d'informations sur la structure du graphe à interroger

Langage SPARQL Update

Le langage SPARQL Update permet de modifier des graphes RDF

Il comporte plusieurs requêtes

- INSERT DATA { **sujet prédicat objet** }
- DELETE DATA { **sujet prédicat objet** }
- INSERT { **template** } WHERE { **template** }
- DELETE { **template** } WHERE { **template** }

Langage SPARQL Update

Le langage SPARQL Update permet de modifier des graphes RDF

Il comporte plusieurs requêtes

- **CREATE GRAPH** < uri >
- **CLEAR GRAPH** < uri >
- **DROP GRAPH** < uri >

Langage SPARQL Update

Le langage SPARQL Update permet de modifier des graphes RDF

Il comporte plusieurs requêtes

- **LOAD** < uri >
- **LOAD** < uri > **INTO GRAPH** < uri >



Publier, Consommer et Visualiser LD

LINKED OPEN DATA

213

213

Publier des données liées

- Pour publier une ressource en tant que donnée liée sur le Web on doit satisfaire plusieurs exigences:
 - Identifier chaque ressource avec une **URI unique et stable** qui doit être **déréférençable** via un protocole HTTP ou HTTPS.
 - une **URI déréférençable** est une URI qui peut être utilisée pour accéder à la ressource qu'elle identifie **sans nécessiter d'autres étapes ou transformations**. En d'autres termes, elle peut être utilisée pour **recupérer la ressource directement**
 - Utiliser des **formats de données ouverts, standardisés** et largement utilisés pour représenter les données, tels que RDF, JSON-LD ou Turtle.

LINKED OPEN DATA

214

214

Publier des données liées

- Utiliser des ontologies et des vocabulaires communs et bien documentés pour **décrire les termes** utilisés dans les données.
- Fournir des **métadonnées précises** pour chaque ressource pour décrire ses **caractéristiques et ses relations** avec d'autres ressources.
- Assurer la **disponibilité et la stabilité** des **URIs déréférençables** pour les **ressources liées**, en maintenant des liens valides entre ces ressources et en évitant les changements d'URI.
- Permettre aux utilisateurs **d'accéder** aux **données liées** via des liens, en fournissant des descriptions de liens (link headers) ou des **redirections HTTP**.

Publier des données liées

- Fournir des liens vers d'autres ressources liées pour faciliter la navigation et la **découverte de données connexes**.
- Assurer l'utilisation interne en production de SPARQL pour garantir que les données sont **stockées correctement** et que les **requêtes peuvent être exécutées efficacement**.
- Assurer l'**accès public à un serveur SPARQL** pour permettre l'interrogation et la récupération de données liées à travers des requêtes standardisées.
- Respecter les bonnes pratiques de sécurité pour protéger les données publiées, telles que la **gestion des autorisations d'accès**, la **validation des entrées utilisateur** et la **protection contre les attaques**.

Publier des données liées

Utilisation des fichiers RDF statiques :

- La façon la plus simple d'utiliser les données liées c'est de créer un fichier statique RDF et de l'enregistrer sur un serveur web.
- Cette technique est utilisée :
 - Quand les fichiers RDF sont créés manuellement généralement utilisés pour publier les fichier personnelle FOAF ou les vocabulaires RDF.
 - Quand les fichiers RDF sont générés par une partie d'un logiciel

Publier des données liées

Utilisation des fichiers RDF statiques

- Lorsqu'on utilise un fichier RDF statique par exemple <http://example.com/people.rdf>, on doit nommer les ressources non-informationnelles décrites dans le fichier en ajoutant un identifiant de fragment à l'URI du fichier : L'identifiant doit être unique au fichier.
- URIs auront ainsi cette forme:
 - <http://example.com/people.rdf#aaaa>
 - <http://example.com/people.rdf#bbbb>
 - <http://example.com/people.rdf#cccc>

Publier des données liées

Utilisation des bases des données relationnelles

- Dans le cas où les données sont stockées dans une base de données **relationnelle** (plutôt que de convertir toutes les données dans un format RDF) il est généralement plus pratique de les laisser telles qu'elles sont et de publier une **vue Linked Data** de cette base de données.
- Une vue Linked Data peut être créée en utilisant des outils tels que D2RQ, RDF Views, OpenLink Virtuoso pour mapper les tables de la base de données à des URI, des propriétés et des classes RDF:
 - Les données peuvent être ainsi facilement accessibles sous forme de données liées sans avoir à dupliquer les données dans un autre format.

LINKED OPEN DATA

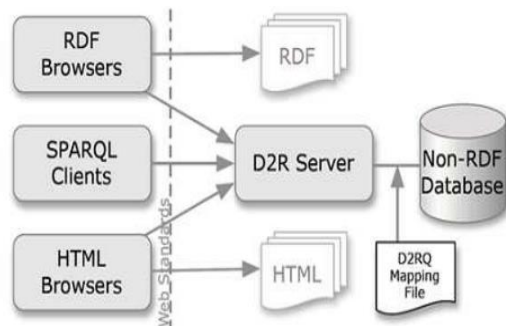
219

219

Publier des données liées

Utilisation des bases des données relationnelles

- Le serveur D2R permet de créer des vues de données liées à partir des bases de données relationnelles.
- Le serveur D2R repose sur une méthode déclarative entre les schèmes de la base de données et les termes RDF.
- D2R crée une vue de données liées sur la base de données cible et fournit pour elle un endpoint SPARQL.
 - Le serveur Berlin DBLP Bibliography Server : publie des données bibliographiques sur des publications scientifiques.
 - Le serveur Hannover DBLP Bibliography Server : publie également des données bibliographiques sur des publications scientifiques.



LINKED OPEN DATA

220

220

Publier des données liées

Autres formats(CSV, Excel, ...)

- Si les informations sont représentées dans des formats tels que CSV, Microsoft Excel, ou BibTEX et on veut les publier sous forme de données liées sur le Web, on suit les étapes suivantes :
 - Convertir les données en RDF.
 - Stocker les données converties dans un répertoire RDF.
 - Le répertoire choisi doit être compatible avec l'interface de linked data pour que ses données soient accessibles sur le web.
 - on peut choisir un répertoire qui fournit un endpoint SPARQL et Pubby comme interface de données liées.

LINKED OPEN DATA

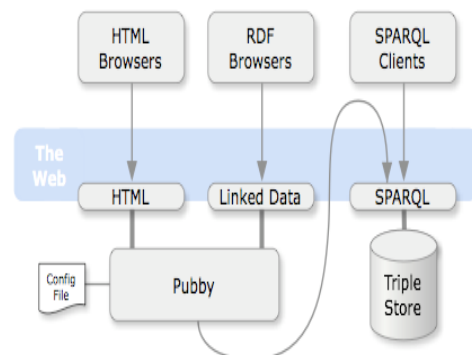
221

221

Publier des données liées

Autres formats(CSV, Excel, ...)

- Pubby est un serveur web open-source qui permet de publier des données liées.
- Il prend en charge plusieurs formats de données tels que RDF/XML, Turtle, N-Triples, JSON-LD et HTML.
- Il permet de publier des données provenant de différentes sources telles que des fichiers RDF, des bases de données relationnelles et des services web RESTful.
- Il fournit également des fonctionnalités de navigation pour parcourir les données publiées et de recherche de ressources à l'aide de critères de recherche spécifiques.



LINKED OPEN DATA

222

222

Publier des données liées

Création des wrappers autour des API

- Plusieurs applications Web ont rendu leurs données disponibles sur le Web via **les API Web**. Qui fournissent **des interfaces d'interrogation** et renvoient des résultats en différents formats tels que XML, JSON,..
- Il existe trois inconvénients pour l'utilisation des APIs web sont :
 - Le contenu d'une API Web ne peut pas être analysé par les moteurs de recherche
 - Les APIs Web ne sont pas accessibles par des navigateurs de données génériques
 - Les Mashup sont utilisés pour quelques sources de données et ne peut être utilisé pour les nouvelles sources de données (. si une source de données n'a pas d'API Web publique, il sera difficile de créer un Mashup qui combine cette source avec d'autres données.)

Publier des données liées

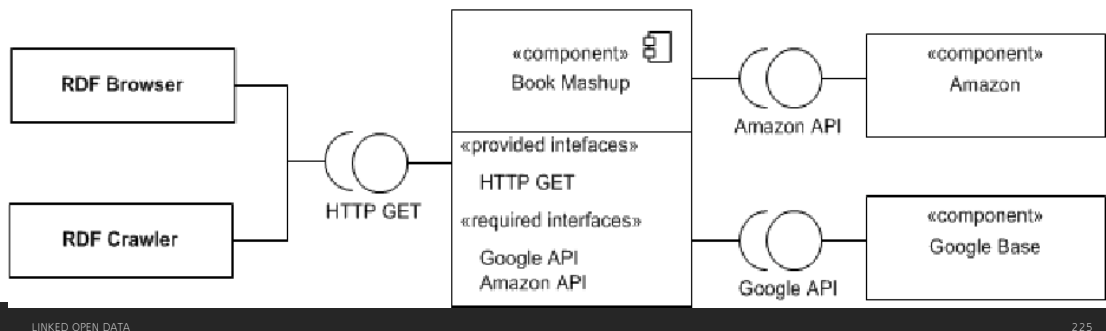
Création des wrappers autour des API

- Pour améliorer le fonctionnement des API, on peut intégrer **Les wrappers de données liées** qui procèdent comme suit :
 - Ils assignent des URIs HTTP aux ressources non-informées disponibles via l'API.
 - Lorsqu'un de ces URIs est dereférencé en demandant l'application/rdf+xml (la demande d'un client pour récupérer les données d'une ressource au format RDF/XML), le wrapper réécrit la requête du client pour correspondre à une requête de l'API.
 - Les résultats des requêtes de l'API sont transformés en RDF et renvoyés au client sous forme de données liées.

Publier des données liées

Création des wrappers autour des API

- Le RDF Book Mashup : application web qui agrège des données provenant de différentes sources (comme Amazon, LibraryThing et OpenLibrary) pour fournir des informations sur des livres sous forme de données liées RDF.

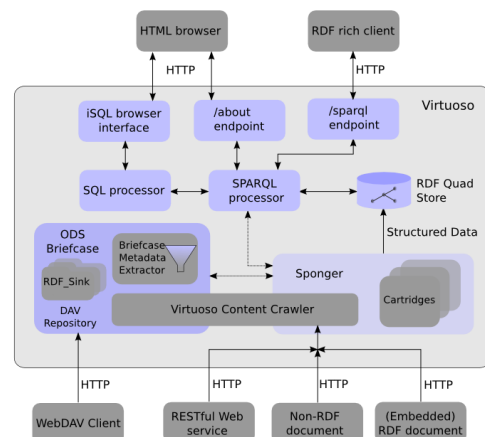


225

Publier des données liées

Création des wrappers autour des API

- Virtuoso Sponger est un composant logiciel du système de gestion de base de données RDF OpenLink Virtuoso.
- Afin de garantir l'intégration de données provenant de différentes sources et leur publication en tant que données liées, il permet de **convertir** automatiquement des données brutes (par exemple, des pages Web HTML) en données RDF en utilisant des règles de mappage prédéfinies ou personnalisées.



226