

# Des métadonnées à la description des ressources

## Les langages du web sémantique

Bernard Vatant  
Mondeca



### ◆ 1 À propos de...

Bruno Bachimont présente en détail, dans le chapitre 6, les processus de dématérialisation et de virtualisation du document engendrés par les nouvelles technologies de l'information, en particulier dans le domaine de l'audiovisuel. Une telle analyse nous conduit à l'idée d'un document virtuel, reconstruit de l'extérieur par ses métadonnées et selon les besoins de l'utilisateur, résultat d'une requête, d'un dialogue avec le système d'information.

Mais si la permanence du document disparaît, demeure ce dont le document parle, son *sujet* pour parler comme les documentalistes, ou autrement dit ce qu'il désigne, son *réfèrent* pour employer le langage de la sémiotique. D'un point de vue pragmatique, le sujet est défini par ce que le document et les métadonnées en disent. On va donc s'intéresser ici à cet ensemble document-métadonnées (que celles-ci soient définies *a priori* ou *a posteriori* par rapport à celui-là) en tant que support de la description d'un sujet. Dans une telle approche, les métadonnées documentaires classiques ne constituent qu'un premier degré, un cas particulier dans un cadre général de *description des choses*. Un tel cadre théorique pouvait se concevoir en principe avant même l'avènement du web et de la numérisation généralisée des systèmes d'information, mais ce sont ces technologies, et les questions techniques et conceptuelles qu'elles ont soulevées, qui ont permis son émergence dans les dix dernières années.

Dans une première partie on décrira l'extension, à travers l'histoire courte et mouvementée du web, de la notion de document à celle de ressource adressable, puis de ressource d'information et enfin de ressource tout court. On

verra que les métadonnées, en tant qu'éléments d'identification et de description, non seulement survivent à ces bouleversements, mais se généralisent à tous les types de ressources, réelles ou virtuelles, physiques ou abstraites, et jouent un rôle de plus en plus critique et central dans l'organisation et la recherche de ces ressources.

On montrera pourquoi le standard RDF<sup>1</sup> du W3C<sup>2</sup> s'impose de plus en plus comme un outil d'unification conceptuelle et technique de cette évolution, unification difficile mais nécessaire. D'abord développé comme simple format standard de métadonnées, RDF étend son champ d'application avec l'extension de la notion de ressource et, de manière récursive, il aide à formaliser cette extension. Il s'impose donc maintenant comme la lingua franca non seulement de l'identification et de la description de toutes les « choses », mais aussi de leur mise en relation, et donc de l'organisation des connaissances.

On verra ensuite comment les différents langages construits sur RDF, qui forment la famille des langages du web sémantique, répondent aux différents aspects fonctionnels de cet univers de métadonnées généralisées et occupent chacun une niche correspondant à des domaines auparavant un peu cloisonnés. Classification, classement, taxonomie, ontologie, index, thésaurus, etc. sont tous susceptibles d'être intégrés dans le cadre général, mais en gardant des spécificités que ces différents langages ou vocabulaires doivent respecter. Chantier difficile, mais qui a le mérite de provoquer un audit général de ces pratiques.

On présentera rapidement le langage SPARQL<sup>3</sup>, qui permet d'interroger les descriptions RDF dans un but de recherche, mais aussi de reconstruction documentaire à partir de métadonnées.

Enfin, on passera en revue quelques tendances actuelles du web dit social-sémantique, où les ressources ne sont plus seulement des objets décrits à des fins d'indexation et de recherche, mais des objets d'échange social. Au-dessus de la couche de métadonnées documentaires, plutôt descriptives et soucieuses d'objectivité, on voit se construire une couche de métadonnées sociales: qui échange quoi avec qui, qui dit quoi de quoi, etc. Cette perspective est sans doute la plus dérangementante pour les professionnels de la documentation dont le métier se fonde sur quelques principes structurants selon lesquels le producteur du document, le documentaliste qui le classe et l'indexe et l'utilisateur qui le cherche sont bien séparés. Le web social remet fortement en question cette médiation.

1 Resource Description Framework : [www.w3.org/RDF](http://www.w3.org/RDF)

2 World Wide Web Consortium : [www.w3.org](http://www.w3.org)

3 SPARQL Query Language for RDF: [www.w3.org/TR/rdf-sparql-query](http://www.w3.org/TR/rdf-sparql-query)

## ◆ 2 Des ressources, et de leur description

### 2.1 Du document à la ressource

Le concept de ressource a émergé durant l'histoire du web à partir de la notion primitive de « document en ligne ». Le premier objectif du web, tel qu'imaginé par son créateur Tim Berners-Lee, était de rendre accessible et de relier entre eux des documents scientifiques, de façon à faciliter leur accès et leur échange par la communauté des chercheurs. Dans ses premiers documents de spécification (1990), le web est donc défini comme un réseau d'objets considérés comme statiques, des fichiers ou documents électroniques, reliés par des liens hypertextes et échangeables *via* le réseau Internet grâce à des protocoles spécifiques tels que HTTP ou FTP.

Les technologies du web ayant rapidement conduit à une grande diversification des types de contenu adressable, la notion de document en ligne a petit à petit fait place dans le vocabulaire et dans son utilisation à la notion de « ressource web ». La définition de ce concept a évolué progressivement pour devenir de plus en plus générale et de plus en plus abstraite, jusqu'à recouvrir, dans les toutes dernières spécifications, « toute chose ou entité susceptible d'être identifiée, nommée, manipulée à travers ses représentations, par quelque moyen que ce soit, sur le web en général ou dans n'importe quel système d'information utilisant les technologies du web. »

Le terme anglais *resource*, que nous traduirons ici par « ressource web » ou simplement « ressource » tout court si le contexte n'est pas ambigu, apparaît de façon indirecte en juin 1994 dans le document RFC 1630 [27]. La notion en elle-même n'est pas directement définie, mais elle est en creux dans les définitions des *universal resource identifiers* (URI), *universal resource locators* (URL) et *universal resource names* (URN), qui permettent d'encoder les noms et les adresses sur Internet.

Une ressource est donc implicitement définie comme quelque chose qui peut être nommé et identifié. Dans l'univers du web statique, l'identification présente deux aspects essentiels mais indissociables, le nommage et l'adressage des « objets du réseau ». Identifier, c'est aussi localiser et atteindre par un protocole spécifique, dans le cas des URL au moins. Les URN (comme, par exemple, les codes ISBN) n'étant pas liées à un protocole, elles échappent à cette ambiguïté mais sont par conséquent moins utilisables dans l'environnement web<sup>4</sup>. De fait

<sup>4</sup> Cependant les identifiants utilisés dans les espaces de noms URN peuvent servir de clé commune à l'échange d'informations sur le web. C'est le cas des codes ISBN encapsulés dans des URI http ou échangés à travers des services. Voir, par exemple: <http://isbn.nu/2070418235> ou [www.amazon.ca/gp/product/2070418235](http://www.amazon.ca/gp/product/2070418235) ou [www.biblio.com/isbn/2070418235.html](http://www.biblio.com/isbn/2070418235.html) pour quelques avatars de *Quatrevingt-treize*.

beaucoup des débats qui suivront à propos des URI portent sur l'utilisation des URL, et singulièrement des URI de type http.

Tout en remplaçant subtilement le qualificatif *universal* par *uniform*, le document RFC 1738 [28], six mois plus tard, utilise plus systématiquement mais toujours sans le définir explicitement le terme ressource pour désigner uniquement *les objets qui peuvent être localisés et atteints à travers le réseau*.

La lecture détaillée de ces documents fondateurs signés du concepteur même du web, Tim Berners-Lee, est intéressante car elle montre bien l'articulation, mais aussi le risque de confusion, entre les aspects déclaratifs (identification et nommage) et fonctionnels (accès et protocoles) de la définition d'une ressource. La mauvaise compréhension de cette dualité, et la confusion qui en résulte entre URI et URL, seront à la source de nombreux débats, aussi bien de nature technique que linguistique, sociale ou philosophique, qui ne sont pas clos.

La première définition explicite du terme *resource* dans son sens le plus général est donnée par RFC 2396 [29], en août 1998 : « Une ressource peut être toute chose qui possède une identité. Des exemples familiers incluent un document électronique, une image, un service (par exemple "le bulletin météo d'aujourd'hui pour Los Angeles"), ou un ensemble d'autres ressources. Certaines ressources ne peuvent pas être "ramenées par le réseau" (*network retrievable*); par exemple les êtres humains, les entreprises, les livres d'une bibliothèque peuvent être aussi considérés comme des ressources. »

Les exemples donnés sont encore limités dans ce texte à des « choses physiques », mais la porte est ouverte à la possibilité de ressources plus abstraites. D'ailleurs, si on y regarde de près, l'exemple du livre de bibliothèque, objet emblématique de la pratique documentaire, pose de nombreuses questions dans ce contexte. Qu'est-ce qui est identifié par l'URI d'un livre? Cet exemplaire particulier, numéroté, qui est dans ma bibliothèque? Faut-il envisager les modifications dans le temps, distinguer les propriétés permanentes de celles qui évoluent, comme le lieu physique de stockage, le rayon de la bibliothèque? Ou bien s'agit-il de l'édition générique de cet exemplaire, défini par son code ISBN (qui est une espèce d'URI)? Dans ce cas mon exemplaire est interchangeable avec tout autre exemplaire de la même édition, et c'est un premier niveau de dématérialisation. Ou bien encore est-ce le livre en général, l'œuvre, indépendamment de tout support physique d'édition ou de collection, voire de langue dans le cas d'un ouvrage traduit? Le texte du livre, indépendamment de sa mise en page? Etc.

La ressource que j'identifie comme un livre peut donc se situer à peu près n'importe où dans ce spectre qui va du rigoureusement matériel jusqu'au concept totalement dématérialisé et virtuel. Le fait que ces différents avatars

puissent être représentés et accessibles sur le web n'arrange rien à l'affaire. On voit donc sur cet exemple typique, presque pathologique, la boîte de Pandore que constitue par sa naïveté le texte ci-dessus.

Par ailleurs on notera la récursivité quasi tautologique de la définition. Une ressource est ce qui est identifiable par une URI, une URI identifie une ressource. Donc, dans la mesure où un concept, aussi abstrait soit-il, est identifié par une URI syntaxiquement correcte, alors ce concept peut être considéré comme une ressource. C'est sur cette possibilité que vont s'appuyer RDF et les langages construits sur cette norme, comme on le verra dans la section suivante.

En janvier 2005, RFC 3986 [30] va entériner explicitement et définitivement cette extension de la définition de ressource: « Les concepts abstraits peuvent être des ressources (par exemple, les opérateurs et opérands d'une égalité mathématique), le type d'une relation (par exemple, "parent" ou "employé"), ou des valeurs numériques (par exemple, zéro, un et l'infini). »

Curieusement, il n'est pas question ici de concepts abstraits beaucoup plus utilisés déjà à cette date que les concepts mathématiques, à savoir les classes ou attributs définis dans une ontologie, les descripteurs d'un thésaurus, les rubriques d'une classification. Pourtant en ce début 2005, les langages d'ontologie RDFS<sup>5</sup> et OWL<sup>6</sup> sont déjà des standards depuis presque un an, et Skos<sup>7</sup>, le langage de représentation des thésaurus et autres vocabulaires structurés, en voie de standardisation. Nous y reviendrons.

## 2.2 RDF comme langage de métadonnées

Publié dans sa première version en 1999 [26] et finalisé en 2004 [22], le Resource Description Framework a pour but, comme son nom l'indique, de décrire les ressources, en d'autres termes d'exprimer de façon standard les métadonnées sous forme de propriétés des ressources. RDF s'appuie sur le constat simple qu'une métadonnée est un couple (propriété, valeur), l'ensemble de ces couples constituant une description de la ressource à laquelle ils s'appliquent. La description RDF d'une ressource est donc un ensemble de triplets (sujet, prédicat, objet) où le sujet est la ressource à décrire, le prédicat une propriété applicable à ce sujet, et l'objet une valeur de cette propriété.

La puissance de RDF tient au fait que non seulement le sujet, mais aussi le prédicat lui-même, sont obligatoirement des ressources identifiées par des URI. L'objet valeur de la propriété peut être une ressource ou une donnée.

<sup>5</sup> RDF Schema : [www.w3.org/TR/rdf-schema](http://www.w3.org/TR/rdf-schema)

<sup>6</sup> Web Ontology Language : [www.w3.org/2004/OWL](http://www.w3.org/2004/OWL)

<sup>7</sup> Simple Knowledge Organisation System : [www.w3.org/2004/02/skos](http://www.w3.org/2004/02/skos)

### 2.2.1 RDF sur un exemple

On va maintenant illustrer les principes de RDF sur un exemple qui nous servira de référence tout au long de ce chapitre. On partira d'une description en langage naturel, telle qu'on peut la trouver dans une encyclopédie, un dictionnaire, ou tout autre ouvrage de référence :

*Quatrevingt-treize est un roman de Victor Hugo paru en 1874, et dont le thème est la Révolution française.*

Une telle description peut facilement (c'est un exercice qu'on peut proposer à l'école élémentaire) être décomposée en propositions simples, chacune exprimant un couple (propriété, valeur) applicable au sujet, ou en d'autres termes une métadonnée et sa valeur :

1. "Quatrevingt-treize est un roman"
2. "Quatrevingt-treize a pour auteur Victor Hugo"
3. "Quatrevingt-treize est paru en 1874"
4. "Quatrevingt-treize a pour thème la Révolution française"

Pour se rapprocher de RDF, on écrira en triplets abstraits de la façon suivante :

1. (*Quatrevingt-treize*, type, roman)
2. (*Quatrevingt-treize*, auteur, Victor Hugo)
3. (*Quatrevingt-treize*, année de parution, 1874)
4. (*Quatrevingt-treize*, thème, La Révolution française)

Nous allons maintenant traduire cette information dans un RDF conforme au standard.

Il faut d'abord que le sujet commun soit identifié par une URI. Une telle URI nous est proposée, par exemple, dans la base de données DBpedia [5], qui offre des descriptions RDF des sujets des articles de Wikipédia, descriptions obtenues à partir de l'information structurée contenue dans ladite encyclopédie en ligne. Notre sujet y est représenté par l'URI <http://dbpedia.org/resource/Ninety-Three>. On verra plus bas l'intérêt d'utiliser une URI http (une URL, pour employer un vocabulaire obsolète mais tenace) et les questions soulevées par cette utilisation. Pour l'instant considérons cette URI simplement comme un identifiant unique, qu'on simplifiera pour la lecture, et selon les conventions usuelles, par `dbpedia:Ninety-Three`.

Le triplet 1 définit un type pour le sujet. La valeur du type, ici « roman », est identifiée dans notre base DBpedia par l'URI <http://dbpedia.org/class/yago/Novel106367879>, en abrégé `yago:Novel106367879`. À noter que les types utilisés par DBpedia sont empruntés à l'ontologie Yago, qui elle-même est dérivée du vocabulaire générique Wordnet, dans lequel 106367879 identifie le concept (ou *synset*) « *novel* ».

La propriété « type » est l'une des rares inscrites en dur dans le vocabulaire RDF, du fait de sa généralité. On écrira donc le triplet 1 de la façon suivante :

```
dbpedia:Ninety-Three    rdf:type                yago:Novel106367879
```

Dans le deuxième triplet, la propriété « auteur » est définie dans le vocabulaire de métadonnées standard Dublin Core [9] par <http://purl.org/dc/terms/creator>. La valeur est également définie par DBpedia [http://dbpedia.org/resource/Victor\\_Hugo](http://dbpedia.org/resource/Victor_Hugo). Avec les mêmes conventions d'écriture que plus haut, on écrira donc :

```
dbpedia:Ninety-Three    dcterms:creator        dbpedia:Victor_Hugo
```

Le troisième triplet est différent des précédents. La valeur de l'objet est une donnée numérique qu'on ne jugera pas nécessaire de représenter par une URI, mais par une date formatée de façon standard. Là encore le Dublin Core fournit le vocabulaire pour la propriété et on écrira :

```
dbpedia:Ninety-Three    dcterms:created "1874"
```

Enfin, le quatrième triplet représente une indexation thématique, pour laquelle DBpedia utilise à juste titre la propriété `skos:subject` empruntée au vocabulaire Skos (nous y reviendrons), la valeur étant la traduction RDF d'une « catégorie » de Wikipédia :

```
dbpedia:Ninety-Three    skos:subject           dbpedia:Category:French_Revolution
```

### 2.2.2 Extensibilité, monde clos et monde ouvert

La structure de RDF présentée ci-dessus est extrêmement simple, générique et extensible. Elle s'appuie sur une construction naturelle de la description des choses. RDF étant un modèle abstrait de données, il peut être exprimé pratiquement dans des syntaxes très variées, stocké dans un tableur, une base de données, un modèle XML. Une description RDF n'est pas prisonnière d'un schéma de métadonnées (en-dehors de celui, très générique, des triplets). À la description précédente, je peux ajouter d'autres éléments complémentaires, ou au contraire ne sélectionner que ceux qui m'intéressent.

RDF est donc un modèle ouvert, conçu pour permettre la réutilisation et l'intégration de vocabulaires définis de façon totalement indépendante. On voit ci-dessus la réutilisation de données de catégories de Wikipédia, de concepts de Wordnet et d'éléments du Dublin Core, tous préexistants à leur migration en RDF. Les éléments de description précédents peuvent être combinés avec d'autres descriptions du même sujet, y compris celles que je ne connais pas encore, utilisant des vocabulaires encore à inventer. Le roman de Victor Hugo peut être traduit dans des langues ou publié dans des supports et formats inconnus aujourd'hui, on pourra toujours enrichir cette description à ce moment sans changer ni invalider mes données primitives.

Une telle extensibilité est fondée sur un principe évident, mais que le monde clos des bases de données pourrait faire oublier : *aucune description n'épuise son référent, ni ne le définit totalement*. Par exemple, les descriptions précédentes sont valides pour mon exemplaire numéroté ou pour une édition particulière du roman, comme pour l'œuvre indépendante de toute publication particulière, qui est certainement la sémantique implicite du sujet de l'article de Wikipédia. Mais en ajoutant d'autres éléments de description, je peux préciser le référent ; dans le monde clos de ma bibliothèque privée, je pourrais par exemple ajouter les triplets suivants, en utilisant mon espace de noms personnel et le vocabulaire que j'ai créés pour sa gestion :

dbpedia:Ninety-Three	bp:acquisEn	"1987"
dbpedia:Ninety-Three	bp:offertPar	bp:Jacques

Il s'agit bien sûr d'un choix très discutable puisqu'en utilisant la même URI je ne fais pas ici de distinction entre le livre en général, tel qu'il est décrit par DBpedia, et mon exemplaire en particulier. C'est néanmoins une solution facile qui me permet d'incorporer à peu de frais les éléments de description glanés sur DBpedia. Tant que ces données restent dans le monde clos de mon système d'information privé, et si je n'ai qu'un exemplaire de *Quatre-vingt-treize* dans ma bibliothèque, cela ne pose pas de problème. Mais si jamais je veux partager ces données, ou si on m'offre un deuxième exemplaire du roman, l'unicité du référent sera détruite par ce changement de contexte.

À propos de l'utilisation des URI http, on reviendra, un peu plus bas ainsi que dans la dernière section, sur les aspects sociaux du web sémantique, sur cette dialectique entre monde clos et monde ouvert. RDF permet a priori « à n'importe qui de dire n'importe quoi sur n'importe quoi ». Cette expressivité sans limites offre beaucoup de flexibilité dans la gestion des données, mais en même temps pose un certain nombre de problèmes quand à la crédibilité, au filtrage, au caractère public ou privé des informations, à la propriété intellectuelle, etc.

On voit bien que les éléments de description personnels ci-dessus ne participent pas du même niveau que les éléments universellement partageables comme l'auteur ou la date de parution, et que je n'ai pas nécessairement envie de les partager. Par contre je pourrais classer le roman sous d'autres catégories que celles proposées par DBpedia, et vouloir partager cette catégorisation à travers une application de *marquage social en ligne*<sup>8</sup>.

<sup>8</sup> Le français n'a pas encore trouvé d'équivalent satisfaisant à l'anglais *social bookmarking*.

## 2.3 RDF comme langage de description généralisé

On a vu que RDF s'appuie sur la définition la plus générale de la notion de ressource en utilisant des URI pour identifier les « ressources abstraites » que sont les types ou les propriétés. De telles ressources sont donc susceptibles elles-mêmes d'être le sujet d'autres triplets. C'est sur cette récursivité que s'appuieront les vocabulaires RDF, comme RDFS, OWL, et Skos qui définissent comme des ressources les classes, propriétés et concepts.

Grâce au vocabulaire RDFS, par exemple, on peut déclarer explicitement qu'une ressource est une propriété ou une classe. Ainsi on trouvera de telles déclarations pour les éléments du vocabulaire RDF Dublin Core [7]. `rdf:property` et `rdfs:Class` ont une sémantique standard définie par le vocabulaire RDFS :

```
dcterms:creator    rdf:type rdf:property
dcterms:Agent     rdf:type rdfs:Class
```

Les deux ressources précédentes peuvent être reliées par une déclaration de co-domaine ou « *range* », qui contraint les valeurs de la propriété « auteur ». « Un auteur est nécessairement un agent » constitue un élément de description de cette propriété « auteur », qui s'exprimera naturellement, de la même façon que la description d'un livre, en utilisant là aussi un élément du vocabulaire RDFS :

```
dcterms:creator    rdfs:range    dcterms:Agent
```

Au niveau supérieur, les langages standard s'auto-décrivent de même par un ensemble de descriptions, quelquefois superbement récursives, comme :

```
rdfs:range         rdf:type         rdf:property
rdfs:range         rdfs:range       rdfs:Class
```

On peut rassembler tous les triplets cités jusque là dans une seule base de données, selon le modèle générique de données RDF. Mais bien sûr tous n'appartiennent pas au même niveau sémantique, ne sont pas gérés au même endroit et n'ont pas la même étendue d'utilisation :

- les triplets de niveau 0 décrivent les instances de base, comme `dbpedia:Victor_Hugo` ;
- les triplets de niveau 1, définis dans les vocabulaires comme le Dublin Core, décrivent les concepts utilisés pour décrire les instances du niveau 0 ;
- les triplets du niveau 2, définis dans les standards, permettent de construire les vocabulaires de niveau 1.

Enfin on peut distinguer aux niveaux 0 et 1 les *référentiels* qui rassemblent des instances contrôlées et publiées par des autorités dans un domaine, par exemple des entités géographiques publiées par l'Insee [21], ou des thésaurus de référence comme le *Thésaurus général multilingue européen de l'environnement GEMET*, publié en Skos [12].

On détaillera, dans la deuxième partie de ce chapitre, la structure et l'expressivité des vocabulaires définis à ces différents niveaux, la manière dont ils s'articulent avec la notion de référentiel, et le rôle que chacun peut jouer dans un environnement de métadonnées généralisées.

## ◆ 3 Questions et réponses

Dans l'introduction précédente, on a présenté RDF comme un modèle de données simple, générique, extensible et indépendant de toute mise en œuvre spécifique. Mais son implémentation technique, en particulier dans l'environnement ouvert et complexe du web, pose un certain nombre de questions techniques et conceptuelles qui ont fait l'objet de débats souvent animés, et qui ont de fait freiné l'adoption du standard en masquant la simplicité du modèle et en le faisant passer pour plus compliqué qu'il ne l'est en réalité. On présentera ici quelques-unes de ces questions, en insistant sur le fait qu'aucune d'entre elles n'est bloquante pour l'adoption du standard, mais que les utilisateurs doivent être conscients de leur existence pour éviter quelques pièges.

### 3.1 Quelle(s) syntaxe(s) pour RDF ?

Comme nous l'avons vu, RDF n'est pas lié à une syntaxe particulière, mais le standard définit en particulier une syntaxe XML pour RDF, et c'est en RDF/XML que sont publiés la plupart des vocabulaires standard (RDFS, OWL, Skos, FOAF, etc.). Mais la syntaxe RDF/XML supporte une grande quantité de variantes, et les rapports entre RDF et XML ont toujours été du type « Je t'aime, moi non plus ». De fait le standard ne définit pas de syntaxe cano- nique pour l'expression XML de RDF, ce qui rend complexe l'intégration de données RDF/XML dans un flux XML à base de schémas, et difficile l'utilisa- tion de transformations XSL par exemple. Cette particularité a fait que, pendant un certain temps, la communauté XML a largement boudé RDF.

Cependant de nombreux outils produisent du RDF dit « *XML friendly* », dont la syntaxe s'appuie sur un schéma stable. Dans un système d'information clos, il faut donc soigneusement choisir les outils RDF pour les interfacer dans une chaîne de production XML.

Par ailleurs d'autres syntaxes sont possibles et utilisées par les logiciels, et toutes sont valides à partir du moment où elles peuvent s'interpréter de façon univoque comme l'expression d'un ensemble de triplets. Il est hors du péri- mètre de ce chapitre de présenter toutes ces syntaxes, mais on peut citer, par exemple, Turtle [33], N3 [20] ou sa variante N-Triples.

Ce qui est à retenir est que l'intégration de RDF dans un système d'information doit absolument s'appuyer sur une interprétation sémantique des triplets. Une erreur commune est de traiter RDF uniquement au niveau syntaxique. Dans ce cas il vaut mieux s'en tenir à des schémas XML...

## 3.2 Pourquoi utiliser des URI http?

L'utilisation d'URI http, pour l'identification de ressources abstraites telles que classes, propriétés ou toute autre espèce de concept, est une pratique courante en RDF, comme les exemples introductifs le montrent. C'est même une pratique recommandée par les standards eux-mêmes, puisque les concepts « de niveau 2 » sont identifiés dans ces standards par de telles URI. L'intérêt d'une telle pratique est de pouvoir attacher, au-delà de l'usage de l'URI comme identifiant du concept, une description de référence à laquelle le protocole http permet d'accéder. Mais dans la pratique, cela pose un certain nombre de questions.

### 3.2.1 Le problème httpRange-14

Quelle forme de réponse le protocole http doit-il renvoyer à une requête utilisant l'URI d'une ressource abstraite, et comment distinguer cette ressource abstraite d'une ressource documentaire (une page web)? Cette question difficile est longtemps restée ouverte sur le bureau du groupe Architecture technique du web au W3C sous le nom devenu célèbre de « problème httpRange-14 ».

Ce groupe a finalement publié en 2005 une réponse d'autorité [15]. Sans rentrer dans les détails techniques dont on donnera un aperçu dans la section suivante, on peut résumer en disant que la réponse à la requête n'est pas définie de façon unique côté serveur mais dépend du dialogue client-serveur par une négociation de contenu qui distingue une réponse pour les humains (description informelle du concept dans un document texte ou html) d'une réponse pour les machines (description formelle en RDF du concept). Cette solution a rencontré un consensus de la communauté du web sémantique, même si quelques voix se sont élevées pour critiquer sa difficulté de mise en œuvre technique ou même son fondement conceptuel. Des bonnes pratiques conformes à cette résolution ont été documentées et mises en œuvre en vraie grandeur, en particulier dans le cadre des initiatives Linked Data [14] dont nous reparlerons en détail plus bas.

### 3.2.2 « Slash », « Hash » et négociation de contenu

La résolution précédente a permis en particulier de traiter la question des URI utilisables, qui peuvent être de deux types, pour représenter une res-

source abstraite. Dans le cas d'un vocabulaire relativement restreint, comme une ontologie, on peut inclure toutes les descriptions dans un seul document, chaque concept étant défini par un identifiant interne au document.

Par exemple, le vocabulaire RDFS dont nous avons cité des éléments plus haut est défini dans le document identifié par (et adressable à) [www.w3.org/2000/01/rdf-schema](http://www.w3.org/2000/01/rdf-schema). Dans ce document on trouve la description RDF des concepts du vocabulaire RDFS, définis par des URI utilisant l'espace de noms [www.w3.org/2000/01/rdf-schema#](http://www.w3.org/2000/01/rdf-schema#), comme par exemple [www.w3.org/2000/01/rdf-schema#Class](http://www.w3.org/2000/01/rdf-schema#Class). Un navigateur « pour les humains » interprétera cette dernière URI comme une ancre dans un document, et ramènera donc la totalité de celui-ci, alors qu'une application capable de lire le RDF extraira la description spécifique du concept (les triplets dont cette ressource est le sujet).

Pour un vocabulaire plus vaste, on aura intérêt à publier un document par concept. C'est ce que fait par exemple DBpedia. Conformément à la résolution de [httpRange-14](#) citée plus haut, une requête http sur l'URI <http://dbpedia.org/resource/Ninety-Three> renvoie une réponse de type « 303 see other » qui, à la fois, indique que l'URI demandée n'identifie pas un document mais une ressource abstraite et redirige vers un document contenant une description de la ressource en question. Suivant le type de contenu demandé par la requête, ce document peut être une page html mise en forme pour une lecture humaine, dans ce cas <http://dbpedia.org/page/Ninety-Three>, ou bien un document RDF pour une utilisation par les applications qui savent le lire, en l'occurrence <http://dbpedia.org/data/Ninety-Three>.

Pour bien comprendre ce mécanisme qui paraît au départ un peu tarabiscoté, on peut utiliser des extensions de navigateur qui permettent de demander au choix l'une ou l'autre des deux représentations<sup>9</sup>.

### 3.3 Comment construire de « bonnes » URI ?

Très liée à la précédente et souvent confondue avec elle, cette question exige beaucoup de soin et d'attention. Il est clair que la définition et la gestion d'un espace de noms pour des URI http nécessitent d'avoir le contrôle sur un nom de domaine et sur le serveur qui l'héberge. Mais l'espace de noms n'est pas seulement un sous-domaine, c'est aussi un contexte dans lequel chaque chose a un nom, et chaque nom correspond à une chose. Par nom, il faut lire bien sûr identifiant. Là encore sans entrer dans le détail, on peut pointer quelques bonnes pratiques.

<sup>9</sup> Par exemple, on peut utiliser l'extension Tabulator pour Firefox : [www.w3.org/2005/ajar/tab](http://www.w3.org/2005/ajar/tab)

### 3.3.1 Les bonnes URI sont stables

« *Cool URIs don't change.* » Cette formule a fait fortune et reste un des principes les plus évidents mais les plus malmenés en pratique dans notre monde de plus en plus mouvant. On lira toujours avec profit le document « historique » du fondateur du web à ce sujet [4]. Une URI est un identifiant. En tant que telle, elle se doit d'avoir une pérennité puisque d'autres utilisateurs vont lui faire confiance. Publier des URI crée donc d'une certaine façon des responsabilités, et on pourrait ici parler d'un principe de développement durable de l'économie des connaissances. Donc, avant de publier et d'utiliser un vocabulaire RDF, il est bon de soigneusement réfléchir à la pérennité des URI définies.

### 3.3.2 Les URI sont des identifiants opaques... en principe

En principe, les URI sont des identifiants utilisés par les machines, et c'est en général une mauvaise pratique de vouloir les rendre à tout prix « lisibles par des humains ». La sémantique d'une URI est dans sa description externe (les triplets dont elle est le sujet), pas dans sa structure interne. Cela dit, dans la pratique, on peut rendre ou non les URI lisibles par des humains. Par exemple, les URI suivantes identifient la ville de Dijon et correspondent à trois pratiques différentes, toutes justifiables et largement utilisées :

<http://dbpedia.org/resource/Dijon>  
[http://rdf.insee.fr/geo/COM\\_21231](http://rdf.insee.fr/geo/COM_21231)  
<http://sws.geonames.org/6453767/>

La première URI utilise la concaténation d'un espace de noms avec une chaîne de caractères qui ressemble à un nom en langage naturel. De fait cette URI est dérivée de l'URI originale de l'article Wikipédia <http://en.wikipedia.org/wiki/Dijon>, elle-même dérivée automatiquement du titre de l'article, selon le (bon) principe des wikis : un sujet, un nom, une page, une URI.

Dans le second cas, on a utilisé des identifiants existant dans une nomenclature (le code géographique de l'Insee) pour construire des URI. Cette pratique est à recommander quand de tels identifiants existent. Elle permet de dialoguer avec des bases de données qui n'ont pas encore migré en RDF, mais utilisent ces identifiants comme clés uniques.

Enfin le dernier exemple correspond à un environnement multilingue, incorporant une grande quantité d'objets nommés et codifiés de toutes sortes de façons. Geonames est un référentiel de plus de six millions d'entités géographiques dans le monde entier, avec leurs noms et leurs codes dans une grande variété de formats et de langues. Dans ce cas, utiliser la clé unique de l'enregistrement dans la base de données pour fabriquer l'URI est la solution la plus efficace. Dans ce cas bien sûr, l'URI est totalement opaque pour les humains.

### 3.4 Qui peut dire quoi sur quoi ?

Une URI étant un identifiant partageable, elle peut être utilisée à volonté pour donner des éléments complémentaires de description. Le contenu informatif fourni par une page web accessible *via* une URI http peut et doit être contrôlé par l'éditeur, et donc le propriétaire du nom de domaine correspondant. Ainsi le contenu accessible en ligne à <http://dbpedia.org/resource/Ninety-Three> est contrôlé par les propriétaires du nom de domaine dbpedia.org. Mais d'autres descriptions de la même ressource peuvent être publiées en dehors du domaine, et échapperont donc au contrôle de son propriétaire. Par exemple, je peux publier sur mon site personnel les éléments de description évoqués ci-dessus qui relèvent de ma bibliothèque privée, mais ces éléments échappent bien sûr au contrôle de DBpedia.

Dans le monde ouvert du web, toutes les descriptions ne sont pas équivalentes, et les systèmes d'agrégation de contenu RDF ou les moteurs de recherche devront en tenir compte et, par exemple, mémoriser la provenance des éléments de description qu'ils agrègent, le niveau de confiance qu'on peut leur attribuer selon cette provenance, le profil d'utilisateur faisant confiance à une source donnée, etc. De même les outils de raisonnement logique s'appuyant sur la sémantique des données devront s'attendre à détecter et gérer les contradictions.

### 3.5 Comment gérer la coréférence ?

Enfin, on ne saurait terminer cette revue sans soulever une question encore largement ouverte. On a vu ci-dessus que, pour une URI donnée, les descriptions pouvaient être multiples, complémentaires, voire contradictoires. Il existe un problème dual. Des éditeurs différents, voire le même éditeur, peuvent avoir publié des URI distinctes pour des ressources considérées *a posteriori* comme identiques, similaires ou coréférentes (c'est-à-dire ayant le même référent). Le langage OWL permet de déclarer l'identité formelle de deux ressources, par la propriété owl:sameAs. Mais il s'agit là d'une sémantique très forte. Par exemple :

```
<http://sws.geonames.org/6453767/> owl:sameAs <http://rdf.insee.fr/geo/COM_21231>
```

Ce triplet signifie que l'entité identifiée par Geonames sous le numéro 6453767 et la commune de code Insee 21231 sont une seule et même entité (ici la commune de Dijon). La sémantique de cette assertion est que tout élément de description de l'une est automatiquement applicable à l'autre. Il s'agit véritablement de deux noms de la même chose, d'une seule et même ressource.

Par contre, si on considère deux éditions différentes de notre roman de référence, avec des codes ISBN différents, et les URI des pages web qui leur sont consacrées, comme par exemple <http://isbn.nu/0685115186/> et <http://isbn.nu/2070418235/> entre mille autres, quelle relation établir entre ces URI et notre URI de référence sur DBpedia? Il est clair pour des humains que toutes ces ressources ont un référent commun, mais qu'il ne s'agit pas de la même ressource. L'expression correcte de cette coréférence, et l'usage fonctionnel que peuvent en faire les systèmes, restent une question ouverte à ce jour<sup>10</sup>. Cependant l'exemple cité plus haut d'utilisation du code ISBN par des applications différentes montre que si la gestion de la coréférence est difficile à intégrer dans la sémantique déclarative de RDF, elle est déjà implémentée de façon très pragmatique par de nombreux systèmes d'information.

## ◆ 4 Des vocabulaires RDF, et de leur emploi

Nous allons maintenant faire un tour d'horizon des différents vocabulaires construits sur RDF et de leur rôle dans un environnement intégré de gestion des métadonnées et des connaissances. Comme on l'a vu plus haut, tout ensemble de descriptions RDF peut se décomposer en trois strates, le niveau 0 des instances ou individus, le niveau 1 des vocabulaires permettant de décrire ces instances, et le niveau 2 des standards permettant de construire et contraindre ces vocabulaires. Si, pour comprendre la logique de description, il est plus simple de commencer par les instances pour monter vers les niveaux supérieurs plus abstraits, la mise en œuvre s'appuie sur les niveaux supérieurs pour construire les niveaux inférieurs, et c'est donc dans cet ordre qu'on va présenter maintenant les choses.

Il est bien sûr hors de question dans un simple chapitre introductif comme celui-ci d'entrer dans les détails techniques de chacun de ces vocabulaires. On s'attachera donc plutôt à montrer l'expressivité spécifique et le périmètre fonctionnel de chacun d'entre eux, toujours en s'appuyant chaque fois que possible sur notre exemple introductif de l'œuvre de Victor Hugo.

### 4.1 Vous avez dit ontologie ?

Avant de présenter ces langages, dissipons quelques malentendus courants sur le mot *ontologie* lui-même dont les origines dans la métaphysique d'Aristote tendraient à le rendre quelque peu suspect aux praticiens des systèmes d'information. On emploie ici ce terme dans un sens qui fait à peu près

<sup>10</sup> L'auteur se bat depuis bon nombre d'années dans les sphères du web sémantique pour attirer l'attention sur l'importance de cette question. Voir <http://universimedia.blogspot.com> (en anglais).

consensus dans la communauté des utilisateurs du web sémantique. D'un point de vue fonctionnel, une ontologie a pour objectif de définir de façon formelle, pour un domaine de connaissances, les concepts qui permettront de décrire « les choses » de façon non ambiguë, et les règles contraignant ces descriptions. Une ontologie doit être compréhensible par les humains et utilisable par les machines pour des tâches diverses comme contrôler des interfaces, filtrer, classier et agréger l'information, le cas échéant déduire de nouvelles informations (inférence).

De façon générale, on considère qu'une ontologie comprend les éléments suivants. Nous utiliserons les exemples de cette « micro-ontologie » pour illustrer les éléments des langages RDFS et OWL dans les sections suivantes.

- ◆ Les types de choses, communément appelés *classes*. Une ontologie destinée à la description des livres aura certainement besoin *a minima* de définir les classes telles que « Document », « Livre » ou « Personne ».
- ◆ Les *propriétés* ou attributs des choses, comme « auteur » ou « date de publication ».
- ◆ Les *contraintes* éventuelles qui relient celles-ci à celles-là, sous forme de règles ou axiomes, comme par exemple:
  1. « Un livre est un document »,
  2. « Quelque chose qui possède un auteur est un document »,
  3. « L'auteur d'un livre est une personne »,
  5. « Un livre a au moins un auteur »,
  6. « Une date de publication est une date »,
  7. « Un document a exactement une date de publication »,
  8. « Une personne n'est pas un document ».

On voit que d'une certaine façon une ontologie est un modèle des « choses qui existent ». À partir de là, on peut avoir différentes approches sur la signification de cette modélisation. Certains auteurs [2] considèrent qu'une ontologie doit être une représentation objective, voire exhaustive, du domaine qu'elle représente, et donc répondre à des critères scientifiques d'évaluation. Selon ce point de vue, une ontologie en tant que modèle du monde aurait le même statut qu'une théorie scientifique, entre autres sa falsifiabilité par l'expérience.

Cette approche peut se justifier pour des ontologies servant de support au raisonnement dans des domaines de connaissance complexes, comme les sciences de la vie. S'il s'agit de mettre en évidence les effets indésirables potentiels d'une nouvelle molécule sur tel récepteur cellulaire ou telle espèce vivante, on conçoit que l'ontologie utilisée doit être aussi proche que possible

de l'état des connaissances en la matière, et on pourra de fait confronter la conformité de tel ou tel concept ou axiome à l'expérimentation.

Mais, dans le cas d'ontologies destinées à faciliter l'accès à l'information et aux connaissances, ce qui est plutôt notre propos ici, on défendra plutôt un point de vue de type social ou juridique. Dans ce cas, pour reprendre la formule de Tom Gruber, « une ontologie est un contrat social [8] ». Les éléments définis dans une ontologie résultent d'un consensus entre utilisateurs sur les choses à décrire et la façon de les décrire. Dans l'exemple ci-dessus, on voit notamment qu'un ouvrage anonyme, une compilation automatique de documents ne seront pas considérés comme des livres si au moins un auteur n'est pas spécifié. On peut ou non être d'accord avec cette règle qui n'a rien d'absolu, ni de vérifiable par expérience.

Le fait d'inclure ou non une telle règle dans l'ontologie est donc plus une question d'accord social que d'objectivité scientifique. L'expression de ce consensus dans un langage formel permet de mettre à plat la terminologie, d'explicitier les désaccords éventuels et les approximations nécessaires, le niveau de granularité désiré, les compromis liés aux contraintes de performance du système ou à l'intelligibilité des interfaces utilisateur, etc.

Bien sûr l'objectif final est en général l'implémentation dans un système d'information, mais le rôle social de la formalisation n'est pas négligeable. On peut construire une ontologie à titre d'audit conceptuel d'un système d'information ou d'une entreprise, dans le but d'y détecter les incohérences, sources de blocages ou de conflits.

Concrètement une bonne ontologie est toujours un compromis entre le « trop précis » impossible à mettre en œuvre techniquement et difficilement réutilisable, et le « trop vague » qui ne satisfait pas la demande utilisateur. Suivant les besoins du système d'information, l'ontologie devra être plus ou moins contrainte, et le choix du langage de formalisation et de son expressivité est primordial. Il est inutile de choisir un langage trop expressif si on n'a besoin que de constructions simples. En particulier le choix entre RDFS et OWL est une question souvent posée. On va donc présenter dans la suite ces deux langages en montrant bien le périmètre de leur expressivité.

## 4.2 RDFS: décrire simplement

RDFS comme OWL est un langage qui s'exprime lui-même en RDF, c'est-à-dire qu'il décrit des concepts standard qui serviront à formaliser des ontologies de domaine. On a présenté plus haut cette notion de récursivité des langages RDF dans l'exemple introductif.

RDFS entend fournir une expressivité minimale pour la définition de classes, de propriétés et d'attachements des propriétés aux classes. Dans notre exemple de la section précédente, RDFS permet d'exprimer tous les concepts et axiomes, à une exception près. On utilisera l'espace de noms fictif `ex:` pour notre ontologie.

RDFS permet la déclaration formelle de classes et de propriétés. À noter que, pour des raisons historiques, le vocabulaire RDFS utilise des concepts définis dans son propre espace de noms, et d'autres définis dans l'espace de noms RDF, ce qui explique des bizarreries comme `rdfs:Class` vs `rdf:Property` :

<code>ex:Document</code>	<code>rdf:type</code>	<code>rdfs:Class</code>
<code>ex:Livre</code>	<code>rdf:type</code>	<code>rdfs:Class</code>
<code>ex:Personne</code>	<code>rdf:type</code>	<code>rdfs:Class</code>
<code>ex:auteur</code>	<code>rdf:type</code>	<code>rdf:Property</code>
<code>ex:pubDate</code>	<code>rdf:type</code>	<code>rdf:Property</code>

RDFS permet la déclaration formelle de hiérarchies de classes et de propriétés.

Les axiomes 1 et 6 [voir page XXX] de notre micro-ontologie s'exprimeront de la façon suivante :

<code>ex:Livre</code>	<code>rdfs:subClassOf</code>	<code>ex:Document</code>
<code>ex:pubDate</code>	<code>rdfs:subPropertyOf</code>	<code>dc:date</code>

La sémantique de ces hiérarchies implique bien sûr l'héritage des types. Un langage de requête qui supporte RDFS devra ramener dans la liste des instances de « Document » toutes les instances de « Livre », et, à une requête utilisant la propriété générique Dublin Core « Date », tenir compte des valeurs de la propriété spécifique « Date de publication ».

De plus, cette hiérarchie supporte aussi l'héritage des déclarations de domaine et co-domaine, utilisant les propriétés respectives `rdfs:domain` et `rdfs:range`. Nos axiomes 2 et 3 vont se traduire respectivement par :

<code>ex:auteur</code>	<code>rdfs:domain</code>	<code>ex:Document</code>
<code>ex:auteur</code>	<code>rdfs:range</code>	<code>ex:Personne</code>

La première contrainte est héritée par toutes les sous-classes de « Document », comme « Livre ». Donc un « livre » peut avoir un auteur. De même pour la seconde contrainte, si « Expert » est une sous-classe de « Personne », un expert peut aussi être un auteur.

Ce type de contrainte peut être utilisé de plusieurs façons par les systèmes :

- contrôler des interfaces de création, d'édition ou d'indexation de ressources. La propriété « Auteur » est utilisable uniquement pour les instances de « Document » et sa valeur sélectionnée dans les instances connues par le système de la classe « Personne » ;
- contrôler l'intégrité de données: une anomalie sera signalée si la valeur de la propriété « Auteur » n'est pas dans la classe « Personne » ;

- classier des ressources ou peupler des bases de connaissances: si X est une valeur de « auteur », alors X est une personne.

Les axiomes précédents concernent des propriétés exprimant des relations entre ressources. RDFS permet également de contraindre les valeurs d'une propriété à être conformes à un type de données, en particulier les types de données définies par la norme XML Schema [37]. Par exemple, pour la date de publication, on pourra contraindre le format qui ne l'est pas par la propriété générique du Dublin Core:

```
ex:pubDate          rdfs:range          xsd:date
```

Selon cette dernière déclaration, la date de publication d'un document doit être conforme au type de donnée « date », à savoir la forme YYYY-MM-DD.

L'expressivité des contraintes de hiérarchie, domaine et co-domaine est suffisante pour bon nombre d'ontologies, et RDFS est vraiment à considérer avant de choisir un langage plus complexe à mettre en œuvre comme OWL. Mais RDFS atteint vite ses limites; il ne permet pas d'exprimer, entre autres:

- les contraintes de cardinalité d'une propriété (minimale ou exacte) comme les axiomes 5 et 7;
- les axiomes faisant usage de la négation, comme la disjonction de classes exprimée dans l'axiome 8;
- les restrictions locales de domaine ou co-domaine. Par exemple, l'auteur d'une « Thèse » (sous-classe de Document) est un « Docteur » (sous-classe de « Personne »);
- les opérations sur les classes, comme la réunion ou l'intersection.

Tous ces besoins d'expressivité sont couverts par le langage OWL.

### 4.3 OWL: décrire finement et raisonner

OWL n'est pas conçu *a priori* comme une extension de RDFS, bien qu'il intègre des éléments de ce vocabulaire comme par exemple `rdfs:subClassOf`, `rdfs:range` et `rdfs:domain`.

OWL est issu d'un compromis entre les exigences de l'intelligence artificielle, notamment des outils fondés sur les *logiques de description* [35], et la compatibilité avec les environnements web et RDF. L'objectif était de créer un langage qui, tout en s'appuyant sur le modèle de données RDF et son expressivité, pouvait permettre de mettre en œuvre les outils de raisonnement logiques. Compromis difficile à trouver, et qui a conduit à définir plusieurs « espèces » de OWL.

Les espèces OWL-Lite et OWL-DL sont conçues pour supporter des inférences en logique de description, le niveau OWL-Lite offrant l'expressivité la plus

faible mais supportant des outils logiques moins complexes à mettre en œuvre. Ces niveaux exigent en particulier la séparation stricte entre classes, propriétés de type « objet », propriétés de type « donnée » et individus (instances de classes). OWL-Full permet toute l'expressivité possible en RDF, mais ne permet pas en général l'usage des outils de raisonnement.

Les logiciels de création et d'édition d'ontologies comme Protégé [32] ou Swoop [31] permettent de valider le niveau d'expressivité ou « espèce » d'une ontologie OWL, et incorporent des outils de raisonnement basés sur le niveau OWL-DL. Divers services en ligne permettent aussi de valider un fichier OWL à partir de son URL [36].

Le niveau OWL-Lite possède une expressivité déjà importante qui, en plus des constructions RDFS, permet, entre autres :

- la déclaration de classes ou de propriétés équivalentes, et d'individus identiques ou différents ;
- la définition de restrictions locales sur une propriété, de type universel (toutes les valeurs de la propriété dans une classe donnée), existentielle (au moins une valeur de la propriété dans une classe) ou de cardinalité 0 ou 1, comme l'exemple ci-dessus « Un livre a au moins un auteur » ;
- la définition de propriétés inverses, transitives, symétriques, fonctionnelles ;
- la définition d'intersections de classes.

Le niveau OWL-DL permet de gérer en plus :

- les constructions basées sur la négation : classes disjointes, complémentaires ;
- les cardinalités de valeur arbitraire, comme « Une personne a exactement deux parents » ;
- les réunions de classes.

Pour les contraintes de type restriction locale ainsi que pour les opérations sur les classes (intersection, réunion, complément), OWL permet l'utilisation de *classes anonymes*, qui ne sont pas nécessairement définies par une URI. Du point de vue RDF, ces classes correspondent à des « nœuds blancs » (*blank nodes*), identifiables localement à l'intérieur d'un fichier RDF-XML, par exemple, mais pas de l'extérieur de ce fichier.

Le niveau OWL-Full possède toute l'expressivité de OWL-DL mais sans la contrainte de garder disjoints les individus et les classes (possibilité de définir des métaclasses), et les propriétés de type « objet » et « donnée ».

En plus du support pour les outils de raisonnement, OWL est prévu pour être utilisé de façon modulaire et distribuée sur le web. Plutôt que de redéfinir tous ses concepts, une ontologie peut « importer » d'autres ontologies, définies par le même éditeur ou disponibles en ligne. Cette possibilité a pour

l'instant laissé beaucoup d'utilisateurs un peu sceptiques. Elle exige un niveau de confiance très fort sur la stabilité et la disponibilité des ontologies importées à la volée sur le web. Par contre la réutilisation de concepts définis dans d'autres vocabulaires est une pratique généralisée et recommandable qui permet une forte interopérabilité des ontologies, et évite la redéfinition de concepts. On peut citer comme exemple de réutilisation l'ontologie bibliographique *Bibliontology* [3], qui reprend des éléments de sept ontologies différentes: géographique, temporelle, événements, adresses, programmes de diffusion, Dublin Core, FOAF.

#### 4.4 Skos: classer, indexer, rechercher

Si RDFS et OWL ont pour vocation de formaliser la description de domaines, ils ne répondent pas directement ni spécifiquement aux fonctionnalités des outils traditionnels de la science documentaire: classer, indexer et rechercher les ressources documentaires, en utilisant des vocabulaires contrôlés et structurés: thésaurus, index, plans de classement, taxonomies.

Pour transférer ces vocabulaires et leur utilisation dans un environnement RDF, ni RDFS ni OWL ne fournissent d'outils spécifiquement adaptés à cet usage. Ces langages d'ontologies permettent de décrire ce qu'est un document, les sous-types de documents et les attributs spécifiques de chacun de ces types, y compris éventuellement le « sujet » du livre. Mais ils ne permettent pas d'exprimer l'organisation des vocabulaires décrivant les sujets.

Pour en revenir à notre exemple initial, ni RDFS ni OWL ne permettent de représenter de façon générique la hiérarchie de « sujets » ou rubriques:

Histoire de France > XVIII<sup>e</sup> siècle en France > Révolution française

On voit bien que représenter cette hiérarchie de concepts comme une hiérarchie de classes au sens OWL ou RDFS reviendrait à définir « Révolution française » comme une sous-classe de « Ressource documentaire », ce qui est pour le moins restrictif et même un peu bizarre. Même si dans les deux cas il s'agit de classer un document par un couple (propriété, valeur), on ne dira pas *Quatrevingt-treize* est un(e) « Révolution française », de la même façon que *Quatrevingt-treize* est un « Roman ». « Révolution française » est un *sujet*, comme dans *sujet de conversation*, ou *sujet de mécontentement*<sup>11</sup>. On peut « parler de la Révolution française », c'est-à-dire utiliser ce sujet comme référent, de toutes sortes de façons, pas seulement pour classer des ressources documentaires.

11 « La France compte trente-six millions de sujets, sans compter les sujets de mécontentement. » H. Rochefort, éditorial de *La Lanterne*, mai 1868.

Si on veut utiliser la hiérarchie de concepts ci-dessus dans un but d'indexation documentaire, sa sémantique fonctionnelle est en gros : si un document a pour sujet, ou autrement dit est classé, indexé sous la rubrique « Révolution française », alors on doit pouvoir le retrouver aussi bien à partir des rubriques plus générales « XVIII<sup>e</sup> siècle en France » et « Histoire de France ». L'autopostage<sup>12</sup>, pour employer le jargon des professionnels, ressemble à un héritage de classes, mais ce n'est pas vraiment la même chose.

La mauvaise compréhension de cette nuance a été et demeure la source de nombreuses discussions et incompréhensions mutuelles entre d'une part les membres de la communauté de l'intelligence artificielle, pas toujours au fait des spécificités de la problématique documentaire, et d'autre part les professionnels de la documentation, assez souvent hermétiques aux subtilités théoriques des logiques de description ; les uns comme les autres, pour des raisons opposées, ne comprenant pas pourquoi on n'emploierait pas une structure de classes RDFS pour représenter un thésaurus.

Skos propose donc un vocabulaire ayant vocation à répondre à cette problématique. Contrairement aux vocabulaires OWL et RDFS qui sont des recommandations W3C depuis début 2004, la standardisation de Skos n'est pas encore achevée à ce jour<sup>13</sup>, mais devrait l'être courant 2008.

Skos est un vocabulaire écrit en RDFS, où la classe générique est `skos:Concept`. Les attributs attachés à un concept sont empruntés au départ au vocabulaire des thésaurus, essentiellement :

- un libellé ou terme préférentiel par langue (`skos:prefLabel`) ;
- des libellés alternatifs ou synonymes (`skos:altLabel`) ;
- des définitions et/ou notes d'applications (`skos:definition`, `skos:scopeNote`) ;
- des concepts plus génériques ou plus spécifiques (`skos:broader`, `skos:narrower`) ;
- des concepts associés, ou reliés de façon non hiérarchique (`skos:related`).

La sémantique de ces attributs, qui constituent le noyau stable de Skos, essaye de rester aussi proche que possible de celle du vocabulaire correspondant dans la pratique documentaire, c'est-à-dire une sémantique relativement « molle ». Par exemple Skos délègue aux implémentations le soin de décider si les relations générique-spécifique doivent supporter ou non l'autopostage.

12 « Procédé permettant d'effectuer automatiquement une indexation complémentaire d'un document ou d'une question par tous les descripteurs appartenant à la même branche de l'arborescence du thésaurus que le descripteur le plus spécifique utilisé lors de l'indexation. L'autopostage générique (vers un niveau supérieur) peut être effectué lors de l'indexation et lors de la recherche. L'autopostage spécifique (vers un niveau inférieur) s'effectue lors de la recherche. » Glossaire en ligne de l'ADBS, [www.adbs.fr/autopostage-16226.htm](http://www.adbs.fr/autopostage-16226.htm)

13 À la date de rédaction de ce chapitre (juin 2008), Skos est au stade de *working draft*, la dernière version datant de janvier 2008.

La publication finale de la recommandation nécessite le règlement de quelques débats à la marge, comme savoir si Skos doit étendre son périmètre au niveau terminologique, par exemple pour exprimer la traduction de synonymes dans les vocabulaires multilingues, et préciser ou non la sémantique des relations *broader-narrower*.

Encore ouvert également reste le problème du confinement des vocabulaires. Skos définit la notion de « schéma de concepts » (*skos:conceptScheme*) qui désigne un ensemble cohérent de concepts, en principe publié et maintenu par un même éditeur. La limite d'un vocabulaire dans un monde ouvert pose quelques problèmes pas totalement résolus en l'état actuel de la spécification, qui laisse la possibilité d'implémentations diverses. Comment gérer les extensions spécifiques d'un vocabulaire générique par un éditeur tiers ?

De même, la cohabitation avec OWL reste à explorer. Comment, par exemple, articuler la notion de schéma de concepts avec la notion de sous-classe de *skos:Concept* ? Skos n'a pas l'expressivité nécessaire pour cela mais n'interdit pas d'intégrer un vocabulaire Skos dans un environnement contrôlé par OWL, ni d'utiliser ce dernier pour définir, en utilisant des classes anonymes, des choses comme : les concepts à intégrer dans le schéma X sont les instances des classes A, B ou C (sous-classes de « Concept »), pour lesquelles la valeur de la propriété Z obéit à telle ou telle contrainte. Par exemple pour des concepts géographiques, le schéma de concepts « Grandes villes d'Europe » regrouperait les instances de la classe « Pays » dont le parent est « Europe », et les instances de « Ville » dont le parent est un de ces pays, et dont la population dépasse un million d'habitants.

Malgré toutes ces difficultés et son caractère de standard encore inachevé, Skos s'impose comme le vocabulaire de référence pour la migration des vocabulaires d'indexation dans l'espace du web sémantique. Il est utilisé comme on l'a vu par DBpedia pour la représentation des catégories de Wikipédia. Dans un registre plus technique où les vocabulaires contrôlés jouent un rôle critique, la communauté astronomique internationale, dans le cadre de l'Alliance internationale des observatoires virtuels [17], a choisi Skos pour la migration et la fédération des vocabulaires et thésaurus astronomiques.

Enfin, il faut bien comprendre Skos non comme une alternative simplifiée à OWL, mais comme une couche complémentaire de représentation. Dans un environnement intégré, le même « concept naturel » peut être susceptible à la fois d'une représentation en tant que classe OWL et en tant que concept Skos (avec deux URI différentes). On pourra ainsi voir cohabiter une classe OWL « Restauration familiale », dont les instances seront des restaurants spécifiques comme « Le Bon Accueil » ou « Chez Léon », et un concept Skos « Restauration familiale » qui indexera tout aussi bien des recueils de recettes de cuisine que les pages web des établissements susdits.

## 4.5 RDFa : intégrer la description dans l'hypertexte

Nous n'avons pas abordé jusqu'ici la question qui était pourtant à la base du développement de RDF, celle des métadonnées utilisables sur le web. RDFS, OWL et Skos permettent de publier des fichiers indépendants, de stocker et d'interroger des données qui correspondent à la description de ressources documentaires ou abstraites.

Mais comment utiliser RDF au quotidien du web, et en particulier comment incorporer des descriptions RDF dans un document HTML ordinaire, de façon à le rendre lisible par les machines comme par les humains? On voudrait incorporer aussi bien les métadonnées du document lui-même, comme, par exemple, une description Dublin Core incluse dans la partie <head> du document HTML, que des métadonnées des « choses dont parle le document », comme des annotations au fil de l'eau, inscrites au plus près du texte lui-même. Par exemple si dans un paragraphe HTML on incorpore notre exemple de référence, le formatage HTML classique serait typiquement le suivant :

```
<p>
  <b>Quatrevingt-Treize</b> est un roman de Victor Hugo, paru
  en 1874, et dont le thème est la Révolution française.
</p>
```

Le langage RDFa permet de considérer ce paragraphe, ainsi que ses différentes sections ou groupes de mots, comme une « microressource », et d'exprimer à l'intérieur du balisage HTML la sémantique du paragraphe ou de chacune des sections du paragraphe, en bref d'incorporer dans l'hypertexte HTML des balises qui expriment la sémantique des choses dont parle le texte. En d'autres termes, il s'agit d'annoter de façon sémantique le texte HTML, sans en altérer la structure pour les navigateurs qui ignoreront ces annotations.

On obtiendra par exemple pour notre paragraphe (les annotations RDFa sont en gras) :

```
<p about="http://dbpedia.org/page/Ninety-Three"
  instanceOf="http://dbpedia.org/class/yago/Novel106367879">
  <span property="dc:title"><b>Quatrevingt-Treize</b></span>
  est un roman de <span property="dc:creator">Victor
  Hugo</span> paru en <span property="dc:date">1874</span>,
  et dont le thème est <div rel="skos:subject"
  resource="http://dbpedia.org/resource/French_Revolution">
  la Révolution Française</div>.
</p>
```

On voit que, dans une telle syntaxe, les éléments RDF sont imbriqués dans le HTML et constituent une couche sémantique interne. Cette couche pourra être utilisée par des moteurs de recherche RDF, des interrogations en SPARQL, et créée soit par des éditeurs humains *via* des interfaces d'annotation permettant

de sélectionner des parties du texte et de les indexer sur du vocabulaire contrôlé, soit par des outils de *text mining* ou d'indexation automatique, ou encore générée à partir de bases de données pour des pages dynamiques.

RDFa n'est encore qu'à l'état d'ébauche sur la table du W3C, mais il suscite déjà beaucoup d'enthousiasme de la part d'une communauté grandissante [24], et les outils [23] qui le supportent se multiplient rapidement, tant pour la création que pour la recherche. En bref, RDFa est généralement cité comme le chaînon manquant qui permettra vraiment l'intégration de RDF dans le web à grande échelle.

## 4.6 SPARQL : interroger le graphe RDF

Un ensemble de triplets RDF, autrement appelé « graphe RDF », qu'il soit stocké dans une base de données, publié comme un vocabulaire ou distribué sur le web, doit pouvoir être interrogé de façon à en exploiter pleinement la sémantique. Un langage d'interrogation ou de requête pour RDF s'appuie sur la structure des triplets et la sémantique des vocabulaires. De nombreux langages de ce type ont été construits depuis le début de RDF, mais SPARQL est d'ores et déjà « le » langage de requête standard.

SPARQL permet d'interroger la structure du graphe sémantique pour sélectionner les ressources répondant à une certaine structure de graphe. Les requêtes les plus simples consistent à filtrer les ressources suivant les valeurs de métadonnées, ce qui n'a rien de très original. « Trouver tous les romans de Victor Hugo » sur DBpedia se traduira par exemple par la requête suivante :

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX yago: <http://dbpedia.org/class/yago/>
PREFIX p: <http://dbpedia.org/property/>

SELECT DISTINCT ?x
WHERE { ?x      rdf:type          yago:Novel106367879.
?x             p:author          dbpedia:Victor_Hugo. }
```

Mais la structure du graphe permet des requêtes utilisant des variables intermédiaires, comme « Trouver les auteurs traitant de la Révolution française ». La base DBpedia ne contient pas de relations directes entre auteurs et sujets traités, mais on peut les trouver indirectement en passant par les ouvrages écrits. La requête s'écrira donc explicitement par « Trouver les auteurs d'ouvrages dont le sujet est la Révolution française ». À noter que l'on n'a pas besoin de connaître explicitement le type de ces ouvrages, ni même leur titre, ni quoi que ce soit d'autre les concernant à part le fait qu'ils existent et servent de jointure entre un auteur et un sujet :

```

PREFIX cat: <http://dbpedia.org/resource/Category:>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX p: <http://dbpedia.org/property/>

SELECT DISTINCT?a
WHERE {?x    skos:subject      cat:French_Revolution.
?x          p:author          ?a.}

```

En exploitant le caractère récursif de RDF, SPARQL permet non seulement de découvrir des éléments de base de connaissance (individus de niveau 0) comme dans les exemples précédents (des romans et leurs auteurs), mais il permet aussi d'interroger le niveau 1 de l'ontologie. Par exemple « Quels types d'ouvrages ont-ils été écrits sur la Révolution française, et par quels auteurs? » :

```

PREFIX cat: <http://dbpedia.org/resource/Category:>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX p: <http://dbpedia.org/property/>

SELECT DISTINCT?c?a
WHERE {?x    skos:subject      cat:French_Revolution.
?x          p:author          ?a.
?x          rdf:type          ?c.}

```

SPARQL autorise également l'expression de requêtes constructives et permet par exemple la ré-indexation des ressources précédentes sur un schéma de métadonnées différent :

```

PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX yago: <http://dbpedia.org/class/yago/>
PREFIX p: <http://dbpedia.org/property/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX pub: <http://mabibliotheque.org/grandpublic/>

CONSTRUCT { ?x dcterms:subject    pub:Romans19eme.
             ?x dcterms:title     ?n.}

WHERE { ?x    rdf:type          yago:Novel106367879.
?x          p:author          dbpedia:Victor_Hugo.
?x          p:titleOrig       ?n.}

```

On alimente avec cette requête une rubrique de taxonomie « Romans XIX<sup>e</sup> », représentée par un concept Skos, avec tous les romans de Victor Hugo trouvés dans DBpedia. Le titre original est ramené dans un champ texte en utilisant le vocabulaire Dublin Core.

SPARQL s'apparente ici à un langage de règles permettant de déduire d'éléments de description existants des éléments complémentaires.

Toutes les requêtes précédentes peuvent être soumises et exécutées en ligne sur la base de données DBpedia *via* son interface SPARQL [6].

## ◆ 5 Bonnes pratiques du web sémantique

Nous avons présenté dans la section précédente les différentes briques de langage du web sémantique. Nous allons maintenant brièvement esquisser quelques principes de mise en œuvre de ces langages dans un environnement intégré de gestion des métadonnées et des connaissances, principes fondés sur une expérience de plusieurs années dans ce domaine avec des cas d'utilisation dans des industries aussi diverses que la documentation légale, le tourisme, l'environnement, l'administration territoriale, l'information médicale, l'information sportive, etc. Chacun de ces domaines a ses spécificités, mais ils ont en commun un noyau de problématiques communes pour lesquelles les langages de la famille RDF offrent des réponses génériques.

### 5.1 Audit des vocabulaires : distinguer le terme du concept

Comme on l'a vu plus haut, la construction d'une ontologie est l'occasion d'un audit des concepts maniés par tous les utilisateurs et systèmes d'information qui auront à s'appuyer sur une sémantique commune. Il est indispensable dans cette étape non pas nécessairement d'aligner les vocabulaires, mais de détecter les synonymies (même concept sous des termes différents) et les homonymies (même terme pour des concepts différents).

Ce dernier point est le plus délicat et le plus intéressant : en effet une terminologie commune passe souvent dans une entreprise pour un accord implicite sur une identité de concepts, ce qui est souvent loin d'être le cas si on y regarde de près. Par exemple dans une entreprise de certification maritime, la typologie des navires dépend du type de réglementation qui s'applique ; les réglementations différentes étant gérées par des services différents de l'entreprise, le *oil tanker* des uns n'est pas forcément le *oil tanker* des autres. Il ne faut pas hésiter dans ce cas à dissocier la sémantique de la terminologie : le même concept pourra apparaître dans des contextes différents sous des libellés différents, et le même terme pourra être utilisé avec des sémantiques différentes pour différents utilisateurs.

Cette distinction entre la terminologie et la sémantique est souvent difficile à comprendre, surtout par les acteurs habitués à une logique terminologique, pour lesquels le même mot doit avoir le même sens quel que soit le contexte. Il faut bien expliquer dans ce cas que le rôle de « clé du concept » est transféré à l'URI, ce qui libère la terminologie.

## 5.2 Réutiliser et relier : la logique « Linked Data »

Comme on doit commencer à le comprendre à la lecture des divers exemples présentés dans les sections précédentes, une partie essentielle de la description des ressources est leur mise en relation par des « propriétés objets ». Du typage par classes ou valeurs d'attributs contrôlés aux relations de base de connaissances, en passant par la catégorisation sur des vocabulaires de type Skos, toute la puissance de RDF en matière d'organisation, de navigation, d'agrégation d'information, de requête et d'inférences est fondée sur cette mise en relation sémantique.

La réutilisation de ressources existantes évite la redondance et le cloisonnement des informations sur « la même chose » représentée  $n$  fois dans des silos ayant chacun leurs structures de données, leurs conventions de nommage, leurs modes d'identification et leurs protocoles d'accès. Dans une migration vers un système utilisant les langages du web sémantique, on devra donc faire l'audit des ressources existantes en détectant les redondances et les répliquions inutiles de référentiels.

Le mouvement Linked Data donne l'exemple de mise en place d'une telle stratégie de réutilisation et de mise en relation à l'échelle des données ouvertes et publiques du web. Mais une stratégie similaire peut être mise en place au niveau du système d'information fermé d'une entreprise, ou d'un système semi-ouvert dialoguant avec les données du web.

## 5.3 Réutiliser les ontologies génériques

Dans la construction d'une ontologie pour un système d'information intégré, même si celui-ci concerne un domaine technique très spécifique, il est bien rare que ne soit pas présentes des informations sur ces « choses » extrêmement génériques que sont les personnes, les lieux, les organisations, les événements, les documents, les outils, les projets, etc., bref tout ce qui répond aux « qui, quoi, où, quand, comment » de l'entreprise.

Des ontologies passe-partout sont réutilisables avec profit, comme bien sûr Dublin Core mais aussi FOAF [11] ou vCard [25] pour la description des personnes et organisations, ou Basic Geo WGS84 [1] pour la géo-localisation. Ces ontologies évitent de réinventer l'eau chaude et permettent une interopérabilité assez générique avec d'autres applications utilisant ces mêmes vocabulaires. Sans être des standards comme OWL ou Skos, ces vocabulaires sont suffisamment partagés et utilisés pour devenir des références de fait.

## 5.4 Réutiliser les données publiées...

En plus des ontologies de référence ci-dessus, de plus en plus de valeurs de référence sont elles aussi publiées soit par les organismes d'autorité, comme, par exemple, le référentiel géographique de l'Insee [21], soit par des prestataires de services qui fournissent un « emballage RDF » de données publiques, comme Geonames [13]. Il est à noter que la frilosité des organismes de référence à publier leurs référentiels en RDF suscite des initiatives privées en ce sens. Par exemple l'auteur publie des URI pour les langages naturels, avec des descriptions reprenant en particulier les codes des diverses normes ISO 639 [18]. Mais ces initiatives devraient être relayées par des organismes d'autorité, selon l'exemple de l'IVOA cité plus haut.

## 5.5 ... et publier ses propres données!

On ne saurait donc trop conseiller ici aux détenteurs et éditeurs de référentiels en ligne, qu'ils soient administrations, communautés d'experts, taxonomistes, etc., de songer à la mise à disposition de ces référentiels en RDF. La migration d'un format structuré à un format RDF n'est pas une tâche d'une grande complexité technique, les bonnes pratiques de publication étant documentées. Le retour sur investissement sera d'autant plus évident que cette pratique se généralisera et que ces vocabulaires seront de plus en plus reliés et intégrés.

Bien sûr les entreprises ont des données confidentielles qu'elles ne veulent ni ne peuvent partager sur le web, mais la part du partageable avec profit est souvent beaucoup plus grande qu'on ne le croit.

À titre d'exemple, l'anecdote suivante. En 2002, Mondeca travaillait pour un grand acteur de l'industrie pharmaceutique dans le cadre d'un vaste projet de fédération des données tout au long du cycle de développement d'un médicament, des premiers tests *in silico* jusqu'à la mise sur le marché. Dans cette industrie hyper-compétitive, la mise au point de nouveaux médicaments est un processus extrêmement sensible, au point que le projet était top confidentiel: le fait même que l'industriel en question appliquât une approche de modélisation sémantique ne devait pas être mentionné. En 2005, on retrouvait les mêmes, assis autour d'une table de la Commission européenne à Bruxelles... avec leurs compétiteurs européens, pour définir le cadre d'un vaste programme européen ayant les mêmes objectifs et la même démarche: le partage et la fédération de données au stade précompétitif du développement des médicaments.

On peut lire sur la présentation du projet pilote InnoMed [16]: « Avec ses 16 grandes entreprises pharmaceutiques coopérant avec 14 universités et 8 PME, ce projet démontre clairement que la collaboration entre plusieurs sociétés pharmaceutiques et les autres parties prenantes est non seulement faisable, mais aussi productive. »

## ◆ 6 Le web social-sémantique est en marche

L'exemple précédent le montre: la réalisation du web sémantique n'est pas seulement une question technique. On a vu que RDF fournit la base d'une pile de langages de représentation et de vocabulaires interopérables. Mais cette boîte à outils techniques ne présente vraiment d'intérêt que si elle s'accompagne d'une démarche sociale d'ouverture et de partage des données et des savoirs. Si chaque silo d'information continue à vivre avec sa propre sémantique et ses propres données protégées derrière des pare-feux, les technologies sémantiques restent applicables mais perdent l'essentiel de leur intérêt, et la plus-value par rapport à une architecture fondée sur des schémas propriétaires, même si elle existe, n'est pas forcément évidente à démontrer.

De ce point de vue, les applications dites du web social, dont le développement a été spectaculaire ces dernières années, offrent un champ privilégié de déploiement des technologies sémantiques. Si les entreprises et les institutions restent frileuses, les individus sont avides de partager tout ce qui peut l'être: fichiers (Flickr, YouTube), favoris (del.icio.us, StumbleUpon), savoirs (Wikipédia), etc. Le nombre d'utilisateurs des services de réseaux sociaux personnels comme Facebook ou professionnels comme LinkedIn, est là pour le montrer. Le nombre de comptes utilisateurs sur les réseaux sociaux en ligne dépasse largement le milliard [19].

Or la tendance qui se fait jour dans tous ces services de partage est d'ajouter de la sémantique pour faciliter la recherche et l'échange de données. C'est sans doute le champ d'applications le plus prometteur à court terme des technologies du web sémantique, en tout cas c'est là qu'elles sont actuellement explorées et testées en vraie grandeur, dans des services comme Twine [34] ou Faviki [10]. Un signe révélateur de cette évolution est que les grands acteurs de la recherche sur le web, restés pendant des années très silencieux au niveau des technologies sémantiques, multiplient les initiatives dans ce domaine.

On conclura donc sur cette note optimiste: le web social-sémantique est en marche, et c'est dans ce domaine que se fera, dans les mois et années qui viennent, les tests du passage à l'échelle des technologies sémantiques.

## ◆ Références

- [1] *Basic geo vocabulary*. [www.w3.org/2003/01/geo](http://www.w3.org/2003/01/geo)
- [2] *Beyond concepts: ontology as reality representation* / B. Smith. 2004. <http://ontology.buffalo.edu/bfo/BeyondConcepts.pdf>
- [3] *Bibliographic ontology specification* / F. Giasson. 2008. <http://biblionto-logy.com>
- [4] *Cool URIs don't change* / T. Berners-Lee. 1998. [www.w3.org/Provider/Style/URI](http://www.w3.org/Provider/Style/URI)
- [5] *DBpedia, querying Wikipedia as a data base*. <http://wiki.dbpedia.org>
- [6] *DBpedia SPARQL endpoint*. <http://dbpedia.org/sparql>
- [7] *Domains and ranges for DCMI properties*. <http://dublincore.org/documents/2007/07/02/domain-range>
- [8] *Every ontology is a treaty* / T. Gruber. 2004. Interview in *AIS SIGSEMIS bulletin*. <http://tomgruber.org/writing/sigsemis-2004.pdf>
- [9] *Expressing Dublin Core metadata using the Resource Description Framework (RDF)*. <http://dublincore.org/documents/dc-rdf>
- [10] *Faviki, social bookmarking tool using DBpedia tags*. [www.faviki.com](http://www.faviki.com)
- [11] *FOAF Vocabulary specification*. <http://xmlns.com/foaf/spec>
- [12] *GEMET Thesaurus*. [www.eionet.europa.eu/gemet/rdf](http://www.eionet.europa.eu/gemet/rdf)
- [13] *Geonames.org ontology and semantic Web services*. [www.geonames.org/ontology](http://www.geonames.org/ontology)
- [14] *How to publish linked data on the Web* / C. Bizer, R. Cyganiak, T. Heath. 2007. [www4.wiwiw.fu-berlin.de/bizer/pub/LinkedDataTutorial](http://www4.wiwiw.fu-berlin.de/bizer/pub/LinkedDataTutorial)
- [15] *httpRange-14: what is the range of the HTTP dereference function?* [www.w3.org/2001/tag/issues.html#httpRange-14](http://www.w3.org/2001/tag/issues.html#httpRange-14)
- [16] *Innovative medicines initiative – Pilot project “InnoMed”*. [http://imi.europa.eu/innomed\\_en.html](http://imi.europa.eu/innomed_en.html)
- [17] *International Virtual Observatory Alliance – Vocabularies in the Virtual Observatory*. [www.ivoa.net/Documents/latest/vocabularies.html](http://www.ivoa.net/Documents/latest/vocabularies.html)
- [18] *Lingvoj.org – Languages of the world*. [www.lingvoj.org](http://www.lingvoj.org)
- [19] *List of social networking websites*. [http://en.wikipedia.org/wiki/List\\_of\\_social\\_networking\\_websites](http://en.wikipedia.org/wiki/List_of_social_networking_websites)

- [20] *Notation 3 – An readable language for data on the Web* / T. Berners-Lee. 1998-2004. [www.w3.org/DesignIssues/Notation3](http://www.w3.org/DesignIssues/Notation3)
- [21] *Publication de données géographiques au format RDF*. INSEE, 2006. <http://rdf.insee.fr/geo>
- [22] *RDF Primer*. W3C Recommendation, 10 February 2004. [www.w3.org/TR/rdf-primer](http://www.w3.org/TR/rdf-primer)
- [23] *RDFa Tools*. <http://rdfa.info/wiki/Tools>
- [24] *RDFa Wiki*. [http://rdfa.info/wiki/RDFa\\_Wiki](http://rdfa.info/wiki/RDFa_Wiki)
- [25] *Representing vCard objects in RDF/XML*. W3C Note, 2001. [www.w3.org/TR/vcard-rdf](http://www.w3.org/TR/vcard-rdf)
- [26] *Resource Description Framework (RDF) - Model and syntax specification*. W3C Recommendation, 22 February 1999. [www.w3.org/TR/1999/REC-rdf-syntax-19990222](http://www.w3.org/TR/1999/REC-rdf-syntax-19990222)
- [27] *RFC 1630: Universal Resource Identifiers in WWW* / T. Berners-Lee. 1994. <http://tools.ietf.org/html/rfc1630>
- [28] *RFC 1738: Uniform Resource Locators* / T. Berners-Lee and al., 1994. <http://tools.ietf.org/html/rfc1738>
- [29] *RFC 2396: Uniform Resource Identifiers (URI)* / T. Berners-Lee and al. 1998. <http://tools.ietf.org/html/rfc2396>
- [30] *RFC 3986: Uniform Resource Identifier (URI)* / T. Berners-Lee and al. 2005. <http://tools.ietf.org/html/rfc3986>
- [31] *SWOOP, a tool for creating, editing, and debugging OWL ontologies*. <http://code.google.com/p/swoop>
- [32] *The Protégé ontology editor and knowledge acquisition system*. <http://protege.stanford.edu>
- [33] *Turtle – Terse RDF triple language* / D. Beckett. 2007. [www.dajobe.org/2004/01/turtle](http://www.dajobe.org/2004/01/turtle)
- [34] “*Twine tie it all together*”. [www.twine.com](http://www.twine.com)
- [35] *Une introduction aux logiques de description* / P. Fournier-Viger. 2005. [www.philippe-fournier-viger.com/description\\_logics/introduction\\_logiques\\_de\\_description.html](http://www.philippe-fournier-viger.com/description_logics/introduction_logiques_de_description.html)
- [36] *WonderWeb OWL ontology validator*. [www.mygrid.org.uk/OWL/Validator](http://www.mygrid.org.uk/OWL/Validator)
- [37] *XML Schema part 2: datatypes second edition*. W3C Recommendation, 2004. [www.w3.org/TR/xmlschema-2](http://www.w3.org/TR/xmlschema-2)