

# Interopérabilité sémantique libérale pour les services et les objets

Maxime Lefrançois

Univ Lyon, MINES Saint-Étienne, CNRS, Laboratoire Hubert Curien UMR 5516,  
F-42023 Saint-Étienne, France  
prenom.nom@emse.fr

**Résumé.** Le Web des données promeut l'utilisation de RDF comme modèle pour les données structurées sur le Web. Cependant, la majorité des services Web consomment et exposent principalement du CSV, JSON, ou XML, des formats non-RDF. Il est peu probable que tous ces services se convertissent un jour aux formats RDF existants. Ceci est d'autant plus vrai dans le contexte du Web des objets, puisque les formats RDF sont pour la plupart textuels alors que les objets contraints préféreront des formats binaires tels que EXI ou CBOR. Dans cet article, nous proposons une approche pour permettre l'interopérabilité sémantique de ces services et objets, tout en leur laissant la liberté d'utiliser leurs formats préférés. Notre approche s'ancre sur les principes de l'architecture du Web et ceux du Web des données liées, et repose sur la définition de Présentation RDF. En supposant qu'une Présentation RDF soit identifiée par une IRI et déréférençable sur le Web, nous montrons comment, avec différents protocoles du Web, un client/serveur peut faire comprendre à l'autre partie comment le contenu d'un message peut être interprété en RDF, ou généré à partir de RDF. Nous nommons ceci la négociation de Présentation RDF. En utilisant ces principes, nous montrons comment les services et objets existants pourraient être rendus interopérables à moindre coût sur le Web Sémantique.

## 1 Introduction

Nous cherchons à faciliter l'accès aux formalismes et outils du Web Sémantique pour les entreprises, services Web, et objets contraints. Un des objectifs clé à atteindre est de permettre à ces différentes entités d'accéder à la signification des messages qu'ils s'échangent : ce qu'on appelle l'*interopérabilité sémantique*. Aujourd'hui sur le Web, les entreprises et services web échangent les données dans une multitude de formats. Le format XML (pas RDF/XML) est encore très présent. Les portails open data préfèrent CSV, et les API web : JSON. Quant aux objets contraints sur le Web des objets (Wilde, 2007; Guinard et Trifa, 2009; Guinard et al., 2010), ils préfèrent des formats légers, potentiellement binaires, tels que EXI (Schneider et al., 2014) ou CBOR (Bormann et Paul, 2014). Dans ce contexte, les formats de données RDF (RDF/XML, Turtle, JSON-LD) ne remplaceront vraisemblablement jamais les formats de données existants. Par contre, le *modèle* de données RDF peut toujours servir de *lingua franca* pour l'interopérabilité sémantique.

## Interopérabilité sémantique libérale pour les services et les objets

Dans cet article, nous repartons des définitions et principes de base du Web et du Web des données, et proposons une approche pour permettre l'interopérabilité sémantique de ces services et objets, tout en leur laissant la liberté d'utiliser leurs formats préférés. Notre approche apporte notamment des solutions aux scénarios suivants :

1. Lorsqu'un serveur répond à un client, il lui envoie son message dans un format arbitraire, tout en lui expliquant comment interpréter le message en RDF. Ceci permet au serveur d'avoir une certaine latitude concernant le choix de la représentation qu'il peut transmettre au client, tout en lui laissant la possibilité d'accéder à la signification du message qu'il lui transmet.
2. A l'inverse, lorsqu'un client cherche à obtenir une représentation d'une ressource, il informe le serveur de la manière dont il préférerait recevoir la représentation de la ressource. Typiquement, un format optimisé pour son usage ou ses contraintes. Le serveur peut alors adapter sa réponse au format qui convient au client, tout en lui laissant la possibilité de l'interpréter correctement.
3. Enfin, un serveur peut mentionner à ses clients potentiels comment ils doivent formuler leurs requêtes pour qu'il sache les interpréter. Il peut là encore demander aux requêtes d'être dans un format optimisé pour son usage.

Le reste de cet article est organisé de la manière suivante. La section 2 construit les fondations de notre approche sur la base des définitions et principes fondamentaux du Web et du Web des données. Nous proposons ensuite une mise en œuvre théorique de cette approche dans la section 3, notamment à l'aide d'une ontologie et de champs d'en-têtes HTTP. La section 4 rapporte enfin une implémentation concrète de notre approche dans une librairie open source basée sur Java Jersey et Apache Jena. Cette librairie permet à un ingénieur qui souhaiterait utiliser notre approche de développer à moindre coût un service HTTP RESTful qui consomme et/ou expose du RDF.

## 2 Bases de l'interopérabilité sémantique sur le Web

Afin d'assurer la compatibilité de nos travaux avec les formalismes du Web, nous proposons d'abord une réécriture d'extraits choisis de document du W3C, qui posent les principes fondamentaux du Web et du Web des données. Ceci nous permet ensuite de construire les fondations de notre approche.

### 2.1 Rappels préliminaires

Cette section propose une synthèse des documents (Jacobs et Walsh, 2004), (Sauer mann et Cyganiak, 2008) et (Cyganiak et al., 2014), orientée pour poser les fondations de notre approche<sup>1</sup>.

Le World Wide Web est un espace d'informations composé d'éléments caractérisés par identifiants globaux, les IRI, sur lequel interagissent des *agents Web* : les *agents utilisateurs* et

---

1. Nous utilisons la terminologie des traductions françaises des documents du W3C - <https://www.w3.org/2003/03/Translations/byLanguage?language=fr>

les *agents logiciels*. Un des principes clé du Web est la séparation du contenu, de la présentation et de l'interaction. Précisons ces concepts.

Les IRI *identifient* de manière unique des *ressources*, qui peuvent être localisées sur le Web, ou en dehors (notamment dans le monde réel). Les ressources dont les caractéristiques essentielles peuvent être empaquetées dans un message sont nommées *ressources informationnelles*. Les ressources localisées sur le Web sont des ressources informationnelles nommées *document Web*. L'information qui décrit l'état d'une ressource informationnelle est ce qu'on appelle le *contenu* de cette ressource.

Une *représentation d'une ressource* est un ensemble de données qui encode le contenu de la ressource (l'information décrivant l'état de cette ressource). On ne peut avoir de représentation que des ressources informationnelles. Le terme *présentation* n'est pas défini précisément. Notre interprétation est qu'il est utilisé pour décrire le lien entre les ressources informationnelles et leurs représentations.

*Déréférencer* une IRI est une action qui a pour but d'accéder à la ressource identifiée. « Accéder » doit être pris au sens large, chaque *protocole d'interaction* pouvant définir plusieurs manières d'accéder à une ressource (ex., HTTP GET : obtenir une représentation de la ressource ; HTTP DELETE : supprimer la ressource). Étant donnée une IRI, les agents web devraient pouvoir obtenir une *description* sur le Web de la ressource identifiée par l'IRI. Cette action est nommée « *chercher* » (*look up*) une IRI, et est plus spécifique que *déréférencer*. C'est ce que permet une suite d'appels HTTP GET avec suivi des redirections. Permettre cette action est important pour pouvoir établir une compréhension partagée de ce que l'IRI identifie. Les agents logiciels devraient obtenir une description en RDF, et les agents utilisateurs devraient obtenir une description adaptée, du HTML par exemple.

Une IRI peut inclure un identifiant de fragment (#). Une IRI sans fragment identifie ce qu'on appelle une *ressource primaire*, et cette IRI augmentée d'un fragment identifie une *ressource secondaire* à cette ressource. Un client Web ne peut déréférencer que des IRI sans fragment.

Avec le protocole d'interaction HTTP, si un serveur répond avec le code de statut 200 OK à un appel à une IRI (qui ne peut pas avoir de fragment), alors : (i) l'IRI identifie un document Web, et (ii) le corps de la réponse HTTP est une représentation qui empaquette les caractéristiques essentielles du document Web, à l'instant où elle a été générée<sup>2</sup>.

Une représentation est une description, mais le contraire n'est pas toujours vrai. Citons deux cas importants pour le Web sémantique, qui permettent à des clients Web d'obtenir des descriptions de ressources du monde réel sans que ces ressources puissent être confondues avec des documents Web :

- Si l'on cherche une IRI HTTP avec fragment, que le serveur répond avec le code de statut 200 OK, alors ce qu'on obtient est une représentation du document Web identifié par l'IRI HTTP sans le fragment. C'est donc une description de la ressource secondaire identifiée par l'IRI cherchée.
- Si l'on cherche une IRI HTTP (avec ou sans fragment), et que dans la suite des redirections on rencontre le code de statut HTTP 303 See Other, alors même si au final on obtient une représentation d'un document Web (un code 200 OK), cette représentation ne sera qu'une description de la ressource identifiée par l'IRI originellement cherchée.

---

2. Cette assertion correspond à la décision prise au bout d'une discussion de plus de 10 ans au W3C : HTTP Range 14 - <https://en.wikipedia.org/wiki/HTTPRange-14>

La  *négociation de contenu*  consiste à fournir plusieurs représentations par l'intermédiaire d'une même IRI, chaque représentation devant par ailleurs être transmise comme un  *flux d'octets* , typé par un type de média internet (Jacobs et Walsh, 2004). Un document Web peut avoir plusieurs représentations avec des types de média internet différents, mais il est également possible qu'il ait plusieurs représentations avec le même type de média internet. HTTP permet notamment la négociation de contenu selon le type de media internet, la langue, ou encore l'encodage de caractères.

Cyganiak et al. (2014, §1.5) utilisent informellement le terme  *source RDF*  pour se référer à une source ou un conteneur de graphes RDF persistant mais changeant au cours du temps. Une source RDF est une ressource dont on peut dire qu'elle a un état qui change au cours du temps. Un instantané de cet état peut être exprimé comme un graphe RDF. Par exemple, n'importe quel document sur le Web qui a une représentation RDF peut être considéré comme une source RDF. Comme toute ressource, les sources RDF peuvent être nommées avec des IRI et par conséquent être décrites dans d'autres graphes RDF.

## 2.2 Formalisation

Notre approche repose sur une extension des définitions de la section précédente, et la formalisation de concepts. Proposons tout d'abord une définition du concept de source RDF basée sur la notion de ressource informationnelle :

**Définition 1** Une source RDF est un document Web dont les caractéristiques essentielles peuvent être empaquetées dans un graphe RDF.

Notre définition de source RDF précise donc la définition informelle de Cyganiak et al. (2014, §1.5), en permettant de confondre ses caractéristiques essentielles et le graphe RDF qu'elle contient.

Soit  $\mathcal{G}$  l'ensemble des graphes RDF,  $\mathcal{O}$  l'ensemble des flux d'octets, et  $\mathcal{M}$  l'ensemble des types de media internet. Nous avons vu qu'une représentation est un flux d'octets typé au moins par un type de média internet, mais qu'il existe d'autres manières de typer ces flux d'octets selon le protocole d'interaction. Nous définissons donc un ensemble abstrait de  *types de flux d'octets*   $\mathcal{T}$ . Chaque type de flux d'octet  $t \in \mathcal{T}$  est au moins associé à un type de média internet  $\text{media}(t)$ , qui décrit le type de média internet du flux d'octet.

**Définition 2** Un flux d'octets typé est une paire  $s = \langle o, t \rangle$  composée d'un flux d'octets  $o \in \mathcal{O}$  et d'un type  $t \in \mathcal{T}$ . L'ensemble des flux d'octets typés est noté  $\mathcal{S}$ .

Une  *présentation RDF*  définit une relation entre des graphes RDF et leurs représentations sous la forme de flux d'octets tous typés de la même manière.

**Définition 3** Une présentation RDF  $p$  est une injection partielle de l'ensemble des graphes RDF  $\mathcal{G}$  vers l'ensemble des flux d'octets typés  $\mathcal{S}$ , de sorte que tous les types des flux d'octets du co-domaine sont égaux. Ce type unique est le type de la présentation RDF  $\text{type}(p) \in \mathcal{T}$ . Le domaine d'une présentation RDF est l'ensemble des graphes RDF qui peuvent être représentés en utilisant cette présentation.

Un graphe RDF peut donc potentiellement avoir plusieurs représentations selon une présentation RDF, mais un flux d'octets typés ne peut être la représentation que d'un graphe RDF. L'ensemble des graphes RDF qui ont une représentation pour la présentation RDF est l'ensemble des graphes RDF *valides*, et l'ensemble des flux d'octets typés qui sont la représentation d'un graphe RDF est l'ensemble des *représentations valides*. Le processus qui permet de passer d'un graphe RDF à une de ses représentations est appelé *abaissement* (*lowering*), alors que celui qui permet de passer d'un flux d'octets typé au graphe RDF qu'il représente est appelé *élévation* (*lifting*)<sup>3</sup>.

Nous proposons ci-dessous une formalisation de ces notions qui se conforme à l'intuition que l'on pourrait avoir de différents formalismes et standards existants, que l'on mentionnera après chaque définition.

Soit  $p \in \mathcal{P}$  une présentation RDF.

**Définition 4** *Un graphe RDF est valide pour  $p$  si et seulement si il appartient au domaine de  $p$ . La règle de validation de  $p$  est une application de  $\mathcal{G}$  vers  $\{\text{vrai}, \text{faux}\}$ , qui associe à tout graphe  $g \in \mathcal{G}$  la valeur vrai si et seulement si  $g$  est valide pour  $p$ .*

Les formalismes SPIN (Knublauch, 2011) ou bien les expressions Shape (Prud'hommeaux et al., 2014; Gayo et al., 2014) permettent de définir des règles de validation pour des graphes RDF.

**Définition 5** *Une règle d'abaissement pour  $p$  est une fonction de  $\mathcal{G}$  vers  $\mathcal{S}$ , qui est injective, et un super-ensemble de  $p$ .*

Les formalismes qui permettent de définir des règles d'abaissement incluent XSPARQL (Akhtar et al., 2008), et STTL alias SPARQL-Template (Corby et Faron-Zucker, 2015). Puisqu'une règle d'abaissement de  $p$  est un super-ensemble de  $p$ , on peut définir plusieurs présentations RDF à partir d'une même règle d'abaissement, en choisissant comment on restreint l'ensemble des graphes RDF valides pour la présentation.

**Définition 6** *Un flux d'octets typé  $t$  est une représentation valide pour  $p$  si et seulement si il appartient au co-domaine de  $p$ . La règle de validation de représentation de  $p$  est l'application de  $\mathcal{S}$  vers  $\{\text{vrai}, \text{faux}\}$ , qui associe à tout flux d'octets typé  $s \in \mathcal{S}$  la valeur vrai si et seulement si  $s$  est une représentation valide pour  $p$ .*

Les formalismes permettant de définir des règles de validation de représentations incluent XML Schema pour les types de média internet basés sur XML, JSON Schema pour ceux basés sur JSON, etc.

La définition de règle d'élévation de  $p$  est plus délicate. Pour chaque flux d'octet typé valide, il s'agit de choisir un unique graphe RDF antécédent par rapport à  $p$ , ce qui est possible grâce à l'axiome du choix. De plus, comme pour les règles d'abaissement, on veut pouvoir définir plusieurs présentations RDF à partir d'une même règle d'élévation. S'en suit la définition suivante :

3. L'emploi des termes *lifting* et *lowering* remonte à la recommandation Semantic Annotation for WSDL and XML Schema - <https://www.w3.org/TR/sawSDL/>

**Définition 7** Une règle d'élévation pour  $p$  est une application injective partielle de  $S$  vers  $G$ , telle que l'image d'un flux d'octet typé  $s$  valide pour  $p$  est l'un des antécédents de  $s$  par rapport à  $p$ .

Les formalismes qui permettent de définir des règles d'élévation sont assez nombreux. Citons le langage de correspondances RML (Dimou et al., 2014), les règles XSPARQL (Akhatar et al., 2008), les règles SPARQL-Generate (Lefrançois et al., 2016, 2017), des métadonnées CSV (Tennison et Kellogg, 2015). Nous considérons également que c'est le cas pour GRDDL<sup>4</sup>.

Nous pouvons donc proposer une interprétation des formats RDF existants selon notre définition de présentation RDF. Par exemple, RDF/XML (Beckett, 2004) est applicable à n'importe quel graphe RDF, et impose aux représentations d'un graphes RDF d'avoir pour type de media internet `application/rdf+xml`. On pourrait donc considérer que RDF/XML définit une unique présentation RDF dont le domaine est l'ensemble de tous les graphes RDF. De cette présentation RDF/XML principale, on peut définir une infinité de sous-présentations RDF, en restreignant l'ensemble des graphes RDF valides. C'est la même chose pour Turtle (Beckett et Berners-Lee, 2008).

Pour JSON-LD (Sporny et al., 2014), qui impose aux représentations de graphes RDF d'avoir le type de media internet `application/json+ld`, il n'existe pas d'unique présentation RDF dont le domaine est l'ensemble de tous les graphes RDF. En effet, de multiples présentations RDF différentes peuvent être définies à l'aide de contextes JSON-LD différents. Pour un contexte JSON-LD donné par contre, JSON-LD définit une unique présentation RDF dont le domaine est l'ensemble des graphes RDF. Comme pour RDF/XML, une infinité de sous-présentations RDF pourraient alors être définies, en restreignant l'ensemble des graphes RDF valides.

On peut donc considérer que le contexte JSON-LD paramètre les présentations RDF qui ont pour type de media internet `application/json+ld`. On peut également utiliser d'autres moyens de paramétrer les présentations RDF, comme justement les règles de validation, élévation, abaissement, ou validation de représentation.

### 2.3 Proposition d'extension des principes du Web des données liées

Avec la définition de nouveaux protocoles d'interaction pour le Web (notamment le protocole CoAP pour le Web des objets (Shelby et al., 2014)), il devient opportun de proposer de préciser et d'étendre les principes du Web des données liées. Nous proposons notamment d'inclure deux règles supplémentaires qui impliquent les concepts de source et présentation RDF définis dans la section 2.2. Notre proposition est donc :

1. utiliser des IRI pour *identifier* les choses ;
2. utiliser des IRI pour lesquelles le protocole d'interaction définit un mécanisme de *recherche* (ce qui est le cas pour HTTP et CoAP) ;

---

4. GRDDL définit un attribut XML qui permet de lier un document XML à une transformation (possiblement XSLT) de XML vers RDF/XML. Si l'on se restreint aux transformations qui transforment tout document XML vers un document RDF/XML valide, alors les règles de transformation GRDDL sont des règles d'élévation.

3. lorsque quelqu'un cherche cette IRI, faire en sorte qu'ils puissent obtenir une *description* de la ressource identifiée par cette IRI ;
4. faire en sorte que la *représentation* renvoyée au client soit celle d'une source RDF ;
5. faire en sorte que le client puisse découvrir la présentation RDF de cette représentation ;
6. inclure des liens vers d'autres IRI, pour qu'il puisse découvrir plus de choses.

### 3 Mise en œuvre d'une l'interopérabilité sémantique libérale sur le Web

Nous discutons dans cette section de la mise en œuvre des principes établis à la section 2.3, dans le but de rendre possible les scénarios décrits en introduction.

#### 3.1 Une ontologie pour décrire les présentations RDF

Comme toute ressource, les présentations RDF et les règles des différents types que nous avons définis à la section 2.2 peuvent être identifiées par des IRI, et être décrites sur le Web en conformité avec les principes énoncés à la section 2.3. Nous proposons une ontologie pour décrire les présentations RDF. Les règles de validation sont attachées aux instances de la classe des *descriptions de graphes*, qui sont donc liées à une ou plusieurs instances de la classe des présentations RDF. Cette ontologie est référencée sur le Linked Open Vocabulary Cloud<sup>5</sup>, et est disponible et documentée à l'URL :

<https://w3id.org/rdfp/>

Le préfixe enregistré pour cet espace de nom est `rdfp`. En guise d'exemple d'utilisation de cette ontologie, le code Turtle ci-dessous décrit une présentations RDF pour le media type `application/xml`. Les règles de validation, élévation, et abaissement pour cette présentation RDF sont nommées par des IRI, qui devraient être déréférencables.

```
@prefix rdfp: <https://w3id.org/rdfp/>.
@base <https://w3id.org/rdfp/example/>.

<graph/xml> a rdfp:Presentation ;
  rdfp:mediaType "application/xml" ;
  rdfp:liftingRule <graph/xml/liftingRule> ;
  rdfp:loweringRule <graph/xml/loweringRule> ;
  rdfp:validatedBy <graph/xml/validationRule> ;
  rdfp:presentationFor [
    a rdfp:GraphDescription ;
    rdfp:validatedBy <validationRule>
  ] .
```

#### 3.2 Découverte directe de présentation RDF pour HTTP

Ce qui peut permettre en pratique la réalisation des scénarios mentionnés en introduction est un « ancrage » (*binding*) sur les protocoles d'interaction du Web. Il s'agit au client et au serveur Web de découvrir d'une façon ou d'une autre des informations sur la présentation RDF utilisée, ou à utiliser. Nous limitons ici le cadre de notre étude de deux manières :

5. <http://lov.okfn.org/dataset/lov/vocabs/rdfp>

## Interopérabilité sémantique libérale pour les services et les objets

- nous ne nous intéressons qu'à la découverte directe, c'est à dire qui survient pendant l'échange entre le client et le serveur. La découverte pourrait être indirecte, notamment via une description d'un objet contraint (*Thing Description*) sur le Web des Objets<sup>6</sup>.
- Nous ne nous intéressons qu'au protocole HTTP. Une mise en œuvre similaire pourrait être proposée pour d'autres protocoles d'interaction, notamment CoAP.

**Présentation RDF utilisée.** La présentation RDF qualifie le lien entre la source RDF et la manière dont elle est représentée. Puisque nous suivons les principes de l'architecture du Web résumés dans la section 2.1, nous souhaitons garder orthogonaux les concepts d'identification et de présentation. Nous soutenons l'idée que la présentation RDF fait partie de ce qui type le flux de données de la représentation, et non pas la ressource elle-même. Il serait opportun d'associer la présentation RDF au type de média internet utilisé via un paramètre de type de média internet, comme par exemple : `application/seas;p="https://w3id.org/rdfp/example/graph/sensor"`. Cependant même si ce paramètre peut être défini pour de nouveaux types de média internet, il n'est pas possible de définir un paramètre global pour tous les types de média internet, ce qui est imposé par la RFC 2045 (Freed et Nathaniel, 1996) :

There are NO globally-meaningful parameters that apply to all media types. Truly global mechanisms are best addressed, in the MIME model, by the definition of additional Content-\* header fields.

Nous introduisons donc un nouveau champ d'en-tête HTTP `Content-Presentation` dans ce but. La valeur de ce champ est une IRI absolue qui identifie une présentation RDF.

En rajoutant de tels champs d'en-tête à chaque réponse 200 OK du serveur, un service existant peut être adapté aux formalismes du Web Sémantique à moindre coût, et faire « comme si » il exposait des sources RDF. Charge alors au client de (i) chercher l'IRI de présentation RDF, (ii) d'interpréter le document RDF obtenu pour trouver l'IRI des règles de validation ou d'élévation associées, (iii) de chercher ces IRI, et (iv) d'utiliser la représentation de ces règles pour interpréter le document en RDF. Cette approche peut être utilisée pour les serveurs contraints également, qui peuvent envoyer des messages dans des formats aussi légers et spécifiques à leur application qu'ils le souhaitent. Tant qu'ils permettent au client de découvrir la présentation RDF utilisée, ces clients peuvent interpréter le message en RDF. Si le client lui-même est contraint, il peut également s'appuyer sur un serveur tiers de confiance pour effectuer la transformation à sa place.

**Négocier la présentation RDF à utiliser.** La négociation de présentation RDF permet au client de préciser ses préférences concernant la présentation RDF qui doit être utilisée pour représenter le graphe RDF dans le corps de la réponse HTTP.

Pour qu'un client puisse transmettre cette information au serveur, une solution est d'introduire un champ d'en-tête HTTP `Accept-Presentation`. La valeur de ce champ est une IRI absolue, qui identifie la présentation RDF que le client aimerait que le serveur utilise.

---

<sup>6</sup> La description des objets et la découverte de ces descriptions d'objets est l'un des axes de travail du groupe d'intérêt W3C pour le Web des objets

En utilisant cet en-tête, un client contraint peut demander à un serveur de lui présenter la réponse dans un format spécifique qu'il peut interpréter. Si le serveur est lui-même contraint, il peut là aussi demander à un serveur tiers de confiance d'effectuer la transformation à sa place.

**Pointer directement à une règle.** Dans certaines situations, il semble déraisonnable de s'attendre à ce qu'un client/serveur : (i) cherche l'IRI de présentation RDF, (ii) interprète le document RDF obtenu pour trouver l'IRI des règles de validation ou d'élévation associées, (iii) cherche ces IRI, et (iv) utilise la représentation de ces règles pour interpréter le document en RDF. Dans les cas simples, qui représentent la plupart des cas, il suffit de connaître l'IRI d'une seule des règles. Nous proposons donc d'introduire des champs d'en-tête HTTP additionnels pour pointer directement vers ces règles. Par exemple :

- l'en-tête HTTP `Content-Lifting-Rule` signifie que le receveur de la représentation peut utiliser la règle d'élévation identifiée par cette IRI pour élever la représentation vers le graphe RDF qui est exactement la ressource identifiée ;
- l'en-tête HTTP `Accept-Lowering-Rule` signifie que le client demande au serveur d'utiliser la règle d'abaissement identifiée par cette IRI pour générer une représentation de la source RDF qu'il pourra interpréter.

## 4 Première implémentation sur Jersey, et démonstration

Nous avons implémenté une librairie Java, `rdfp-jersey-server`<sup>7</sup>, pour démontrer l'utilisation des présentations RDF et de la négociation de présentation RDF. Cette librairie est une extension de Java Jersey<sup>8</sup>, et permet à un ingénieur qui souhaiterait utiliser notre approche de développer à moindre coût un service HTTP RESTful qui consomme et/ou expose du RDF. Elle masque tous les mécanismes de présentation RDF et de négociation des présentation RDF, et simplifie la tâche du développeur final qui peut se concentrer sur la manipulation de graphes RDF comme des *modèles* Apache Jena<sup>9</sup>. La version actuelle (1.2) de `rdfp-jersey-server` est disponible sur Maven central et implémente les règles d'élévation SPARQL-Generate, et les règles d'abaissement STTL. La page <https://w3id.org/rdfp/demonstration.html> décrit une démonstration de cette librairie sur une source RDF identifiée par :

`http://ci.emse.fr/rdfp/example`

### 4.1 Consommer du RDF présenté arbitrairement

`rdfp-jersey-server` utilise les champ d'en-tête HTTP, les principes du Web des données liées, et l'ontologie des présentations RDF, pour élever le corps de la requête HTTP en un graphe RDF lorsque c'est possible. Pour l'instant, les champs d'en-tête utilisés sont `Content-Type` et `Content-Presentation`. Toute la complexité est masquée au développeur final, qui peut directement utiliser un paramètre de type Apache Jena Model dans une méthode « ressource » Jersey :

---

7. <http://w3id.org/rdfp/get-started.html>

8. <https://jersey.java.net/>

9. <http://jena.apache.org/>

## Interopérabilité sémantique libérale pour les services et les objets

```
@POST
public Response doPost(@GraphDescription("https://w3id.org/rdfp/example/graph") Model
    model) {
    ...
}
```

L'annotation `@GraphDescription` informe la librairie `rdfp-jersey-server` de la description de graphe (une instance de `rdfp:GraphDescription`) à laquelle la source RDF est conforme. Si le client ne spécifie pas le champ d'en-tête `Content-Presentation`, alors `rdfp-jersey-server` cherche cette IRI pour trouver la présentation qu'elle peut appliquer pour élever la représentation reçue vers le graphe RDF correspondant. Dans le cas présent, la description de graphe déclare deux présentations RDF, une en `application/xml`, et une en `application/json` :

```
<https://w3id.org/rdfp/example/graph> a rdfp:GraphDescription ;
  rdfp:presentedBy <https://w3id.org/rdfp/example/graph/xml> ;
  rdfp:presentedBy <https://w3id.org/rdfp/example/graph/json> .
```

### 4.2 Négocier une présentation RDF

`rdfp-jersey-server` utilise la description de la présentation RDF pour négocier une présentation RDF du graphe RDF dont elle souhaite envoyer une représentation au client. Elle utilise pour l'instant les champs d'en-tête HTTP `Accept` et `Accept-Presentation`. Toute la complexité est masquée au développeur final, qui peut faire en sorte que la méthode « ressource » Jersey renvoie directement un objet de type Apache Jena Model :

```
@GET
@GraphDescription("https://w3id.org/rdfp/example/graph")
public Model doGet() {
    Model model = ...
    return model;
}
```

L'annotation `@GraphDescription` informe `rdfp-jersey-server` de la description de graphe à laquelle la source RDF est conforme. Si le client ne demande pas de présentation RDF spécifique à l'aide du champ d'en-tête `Accept-Presentation`, alors `rdfp-jersey-server` cherche cette IRI pour trouver une présentation RDF pour abaisser la sortie vers une représentation.

## 5 Conclusion

Dans cet article, nous avons étendu et formalisé certaines définitions des documents du W3C qui posent les fondamentaux du Web. Cette formalisation nous a permis de définir la notion de source RDF et de présentation RDF, ainsi que des règles de validation, élévation, abaissement, et validation de représentation. Ces notions permettent de catégoriser des formalismes et standards existants, qui en retour peuvent être utilisés pour répondre à la question principale de cet article :

*Comment permettre aux agents Web d'être interopérable sémantiquement, tout en leur laissant la liberté d'utiliser leurs formats préférés ?*

Nous avons défini une ontologie pour décrire les présentations RDF, ainsi qu'une manière de mettre en œuvre notre approche pour la découverte directe et la négociation directe de présentation RDF pour le protocole d'interaction HTTP. Parmi les directions possibles de travaux

futurs, on peut donc citer la découverte et la négociation indirecte, la mise en œuvre sur d'autres protocoles, mais également l'étude de l'impact sur les performances (nombre d'appels et taille des messages) que notre approche peut avoir pour une interopérabilité sémantique libérale.

Nous avons décrit une première implémentation open-source au dessus de Apache Jena et Java Jersey. Cette implémentation se limite à HTTP, aux solutions de mise en œuvre décrites dans la section 3, et n'utilise qu'un formalisme d'élévation et un formalisme d'abaissement. D'autres implémentations, en particulier sur des objets contraints, sont en cours de réalisation.

Notre approche peut être implémentée dans des objets contraints, et dans des « serveurs de traduction », formant ainsi un écosystème global d'objets et de services utilisant des formats différents, mais conservant leur interopérabilité sémantique.

## Remerciements

Ce travail a été partiellement financé par le projet ITEA2 12004 Smart Energy Aware Systems (SEAS), ainsi que par une convention bilatérale de recherche avec ENGIE R&D.

## Références

- Akhtar, W., J. Kopecký, T. Krennwallner, et A. Polleres (2008). XSPARQL : Traveling between the XML and RDF worlds—and avoiding the XSLT pilgrimage. In *European Semantic Web Conference*, pp. 432–447. Springer.
- Beckett, D. (2004). RDF/XML Syntax Specification (Revised) W3C Recommendation 10 February 2004. W3C Recommendation, W3C.
- Beckett, D. et T. Berners-Lee (2008). Turtle - Terse RDF Triple Language, W3C Team Submission 14 January 2008. W3C team submission, W3C.
- Bormann, C. et H. Paul (2014). Concise Binary Object Representation (CBOR). IETF RFC 7049.
- Corby, O. et C. Faron-Zucker (2015). Sttl : A sparql-based transformation language for rdf. In *11th International Conference on Web Information Systems and Technologies*.
- Cyganiak, R., D. Wood, et M. Lanthaler (2014). RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation 25 February 2014. W3C Recommendation, W3C.
- Dimou, A., M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, et R. Van de Walle (2014). Rml : A generic language for integrated rdf mappings of heterogeneous data. In *LDOW*.
- Freed, N. et B. Nathaniel (1996). Multipurpose Internet Mail Extensions (MIME) Part One : Format of Internet Message Bodies. IETF RFC 2045.
- Gayo, J. E. L., E. Prud'hommeaux, H. R. Solbrig, et J. M. Á. Rodríguez (2014). Validating and Describing Linked Data Portals using RDF Shape Expressions. In *Workshop on Linked Data Quality, SEMANTICS*. Citeseer.
- Guinard, D. et V. Trifa (2009). Towards the web of things : Web mashups for embedded devices. In *Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences)*, Madrid, Spain.

- Guinard, D., V. Trifa, et E. Wilde (2010). A resource oriented architecture for the web of things. In *Internet of Things (IOT), 2010*, pp. 1–8. IEEE.
- Jacobs, I. et N. Walsh (2004). Architecture of the World Wide Web, Volume One, W3C Recommendation 15 December 2004. W3C Recommendation, W3C.
- Knublauch, H. (2011). SPIN-modeling vocabulary. *W3C Member Submission 22*.
- Lefrançois, M., A. Zimmermann, et N. Bakerally (2016). Flexible RDF generation from RDF and heterogeneous data sources with SPARQL-Generate. In *Proceedings of the 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW'16)*, Bologna, Italy.
- Lefrançois, M., A. Zimmermann, et N. Bakerally (2017). Génération de RDF à partir de sources de données aux formats hétérogènes. In *Actes de la 17ème conférence Extraction et Gestion des Connaissances (EGC'17)*, Grenoble, France.
- Prud'hommeaux, E., J. E. Labra Gayo, et H. Solbrig (2014). Shape expressions : an RDF validation and transformation language. In *Proceedings of the 10th International Conference on Semantic Systems*, pp. 32–40. ACM.
- Sauermaun, L. et R. Cyganiak (2008). Cool URIs for the Semantic Web. W3C Note, W3C.
- Schneider, J., T. Kamiya, D. Peinter, et R. Kyusakov (2014). Efficient XML Interchange (EXI) Format 1.0 (Second Edition). W3C Recommendation, W3C.
- Shelby, Z., K. Hartke, et C. Bormann (2014). The constrained application protocol (CoAP). IETF RFC 7252.
- Sporny, M., G. Kellogg, et M. Lanthaler (2014). A JSON-based Serialization for Linked Data. W3C Recommendation, W3C.
- Tennison, J. et G. Kellogg (2015). Model for Tabular Data and Metadata on the Web. W3C Recommendation, W3C.
- Wilde, E. (2007). Putting things to REST. *School of Information*.

## Summary

RDF aims at being the universal abstract data model for structured data on the Web. However, the vast majority of web services consume and expose non-RDF data, and it is unlikely that all these services be converted to RDF one day. This is especially true in the Web of Things, as most RDF formats are textual while constrained devices prefer to consume and expose data in binary formats. In this paper, we propose an approach to make these services and things reach semantic interoperability, while letting them the freedom to use their preferred formats. Our approach is rooted in the Web architecture principles and the Web of linked data principles, and relies on the definition of RDF presentations. Supposing a RDF presentation is identified by a IRI and dereferenceable on the Web, we discuss what it takes for a client/server to be able to discover how a message content can be interpreted as RDF, generated from RDF, or validated. We propose practical solutions to direct RDF presentation discovery and negotiation. Using these principles, we show how existing services and things can be made interoperable at low cost on the Semantic Web.