How my actuator can understand what your sensor says: Revisiting the Web's Architectural and Linked Data Principles for Legacy Services and the Web of Things

Maxime Lefrançois

How my actuator can understand what your sensor says: Revisiting the Web's Architectural and Linked Data Principles for Legacy Services and the Web of Things

Maxime Lefrançois

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

Abstract. 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 for sensors and other devices in the Web of Things, as most RDF formats are verbose while constrained devices prefer to consume and expose data in concise 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's architectural principles and the linked data principles, and relies on the definition of RDF presentations, which describe the link between RDF graphs and their representations. We introduce the RDF Presentation ontology (RDFP) that can be used to model inputs and outputs of procedures of the W3C and OGC joint SOSA/SSN standard ontology, and inputData and outputData of interaction patterns of things in the W3C WoT Thing Description ontology. We then propose practical solutions for web agents to be able to discover how a message content can be interpreted as RDF, generated from RDF, or validated, with different Web interaction protocols.

1 Introduction

The Web today sees companies and web services exchanging data in various and multiple formats. XML (not RDF/XML) is still very used, CSV is commonly used on Open data portals, APIs often use JSON. Constrained devices on the Web of Things [30,13,14] tend to use concise formats, potentially binary such as EXI [24] or CBOR [5]. This raises the problem of *semantic interoperability* at the data level, i.e., how services and devices can get the meaning of the messages they exchange. RDF has the potential of being the universal abstract data model for structured data on the Web, which would be a first step towards enabling *semantic interoperability* on the Web. Yet, in the current context RDF data formats (RDF/XML, Turtle, JSON-LD) will probably never replace the existing ones. However, the RDF data *model* can still be of use as *lingua franca* for semantic interoperability.

In this paper we are investigating the issue of lowering the required effort for different services and devices to reach *semantic interoperability* at the data level using Semantic Web models, without being too prescriptive on the format they should use. In

particular, we want to avoid complex compromises such trying to stick to JSON-LD, and maintain an ontology, a JSON-LD context, and a JSON schema in parallel.

Studying semantic interoperability at the data level on the Web obviously involves some modeling of the communication between heterogeneous agents on the Web. As a first approximation, let us describe such communication as follows: a *sender* wants to send some content (e.g., the state of a resource) to a *receiver*. It first generates a representation of that content, transmits the so-formed message via the Web to the receiver, that then decodes it and gets some understanding of the original content. For this decoded message to be equivalent to the original content, (1) all of the essential characteristics of the content must be encoded in the message, (2) the encoding and the decoding phase must be symmetric, and (3) the message must not be altered in the transmission medium which is the Web. In such a situation, we say the two Web agents reached a *perfect understanding*.

We commit ourselves to develop an approach that conforms to, and leverages, the Web's architectural principles and the linked data principles. As Web's foundations can be considered domain knowledge like any other, we propose to adopt a rigorous knowledge engineering methodology to answer our research question. We therefore apply the following steps, according to which this paper is structured.

Step 1, analyze the domain. Section 2 reviews three core specifications of the Web [16,23,8], recalls the definition of important concepts, harmonizes or precises them when necessary. The concept of *presentation* that is used but not defined in these specifications is crucial for our work, so we propose some definition for it, which to the best of our knowledge has not be done before. Then we propose some formalization for the concepts of RDF Presentation, Lifting, Lowering, and Validation. We conclude with a proposal to precise and extend the Linked Data principles with new rules involving the concepts of RDF Source and RDF Presentation. This section is an important contribution of this work.

Step 2, develop scenarios. Section 3 depicts three scenarios where agents communicate on the Web, all of which involve some adaptation of one or the other party to reach perfect understanding.

Step 3, extract competency questions. Section 4 lists competency questions that we derive from the scenario to define the scope of the ontology.

Step 5, develop the ontology. Section 5 provides ontological representation for the core concepts formally defined in Step 1, that gravitate around the concept of *RDF presentation*. The result of this step is the *RDF Presentation (RDFP) ontology*, along with simple alignments to the OGC and W3C SOSA/SSN ontology [15], and to the W3C TD ontology [17] that show how it can be used in practice. RDFP is the main contribution of this paper, and is used in the SOSA/SSN specification.

Step 6, qualitatively validate the ontology by showing how it answers the competency questions and enable the scenarios. This step is partly led throughout Section 5, and Section 6 shows how the RDF Presentation ontology and the concepts it defines can be used with some of the Web interaction protocols to answer the complementary set of competency questions from Section 4, and enable scenarios from Section 3. Namely, how can a web agent discover the information about the RDF presentation that is used by its interlocutor, either directly, or indirectly.

2 Conceptualization of the Web's and Web of Data's foundations

In this section, we prepare the foundations of the RDF Presentation ontology reviewing three core specifications of the Web and the Web of Data. In Section 2.1 we recall and harmonize the core terms that are relevant for our work, we formalize them in Section 2.2, then we distill the discussion in Section 2.3 through a set of principles that extend the already well-known Linked Data principles [4].

2.1 Precising the Terminology of the Domain

We want to ensure that our work is compatible with the Web formalisms. In this section we select quotes from [16], [23] and [8], that define the fundamental principles of the Web and of the Web of Data. We propose a definition for the concept of *presentation* that is used but not defined in these specifications and is crucial for our work. We also propose definitions for three additional concepts: *RDF presentation*, *RDF lifting*, and *RDF lowering*.

> The World Wide Web is an information space in which the items of interest, referred to as resources, are identified by global identifiers called [International Resource Identifiers (IRI)][1], [and onto which *Web agents* interact: *user agents* and *software agents*.][16, §1.1]

One of the Web's first key points is the separation of content, presentation and interaction [16, §4.3]. We go further in those concepts in what follows. The IRIs *identify resources* that can be localized on the Web, or outside (especially in the real world) [16, §3.1].

> Resources whose essential characteristics can be conveyed in a message are called *information resources*. [16, §2.2]

Resources that are localized on the Web are information resources called *Web documents* [23, §3.1]. The information describing an information resource state is what we call the *content* of that resource.

> A *representation of a resource* is some data that encodes the content of that resource [(i.e., the information describing an information resource state)]. [16, §3.2]

One can therefore only get representations of information resources. The term *presentation* is not precisely defined in [16]. We propose an informal definition here:

Presentation A *presentation* describes an homogeneous relationship between a set of information resources and their representations.

From [16, §3.1], we define *dereferencing* an IRI as an action aiming to access the identified resource. "Access" must be understood in its broad sense, as each *interaction protocol* can define several ways to access resources (e.g., HTTP GET: get a representation of the resource; HTTP DELETE: delete the resource).

[1] We replaced "URI" by "IRI".

> Given only an IRI, [user agents] should be able to retrieve a [*representation* that describes] the resource identified by the IRI from the Web. [23, §3]

This action is also known as *looking up* an IRI, and is thus more specific than the more general action of *dereferencing* an IRI. The look-up process can involve multiple redirections. An HTTP client, for instance, might have to perform additional HTTP GET requests, one for each encountered redirection, to retrieve a description of the resource identified by the IRI.

> Such a look-up mechanism is important to establish shared understanding of what a [IRI] identifies. Machines should get RDF data and humans should get a readable representation, such as HTML. [23, §3]

We synthesize [16, §2.6] as follows. An IRI can contain a fragment identifier (#). An IRI without fragment identifies a *primary resource* and that IRI augmented with a fragment identifies a *secondary resource* of this resource. A web client can dereference only IRIs without fragment.

With the HTTP interaction protocol, if a server responds with status code 200 OK to a GET at some IRI (that cannot have a fragment), then: (i) the IRI identifies a web document [23, §3.1], and (ii) the HTTP response body is a representation that conveys the web document's essential characteristics at the time it was generated.[2]

Let us stress the fact that a representation is a description, but the opposite is not always true. We can cite two important cases for the Semantic Web that allow web clients to get descriptions of real world resources, while preventing these resources be confused with web documents [23]:

(a) If one looks up an HTTP IRI with fragment, and if the server responds with status code 200 OK, then what one retrieves is a representation of the web document identified by the HTTP IRI without the fragment. It is hence a description of the secondary resource identified by the IRI one was looking up.
(b) If one looks up an HTTP IRI (with or without fragment), and if status code HTTP 303 See Other is encountered in the series of redirections, then even if one ends up retrieving a representation of a web document (status code 200 OK), that representation can only be a description of the resource identified by the IRI one was looking up.

> *Content negotiation* refers to the practice of making available multiple representations via the same [IRI]. [16, §3.2.2]

> Representations should be transmitted as *octet streams* typed by Internet media types. [16, §3.2]

A web document can have several representations with different Internet media types, but it can also have several representations with the same one. For example HTTP allows for content negotiation according to the Internet media type, the language, and the character encoding, among other.

[2] That statement corresponds to the consensus reached after more than 10 years of discussion at the W3C: HTTP Range 14 - https://en.wikipedia.org/wiki/HTTPRange-14

> [[8]] informally use term *RDF source* to refer to a persistent yet mutable source or container of RDF graphs. An RDF source is a resource that may be said to have a state that can change over time. A snapshot of the state can be expressed as an RDF graph. For example, any web document that has an RDF-bearing representation may be considered an RDF source. Like all resources, RDF sources may be named with IRIs and therefore described in other RDF graphs [8, §1.5].

We propose to precise the definition of RDF sources, based on the notion of information resource.

RDF source A *RDF source* is a web document whose essential characteristics can be conveyed in an RDF graph.

Doing so, our definition of RDF sources formalizes the definition of [8, §1.5], allowing to identify the essential characteristics of the RDF source with the RDF graph it contains. We can now propose an informal definition the concept of *RDF presentation*.

RDF Presentation An *RDF presentation* describes an homogeneous relationship between a set of RDF graphs and their representations.

Next section formalizes the notion of RDF presentations.

2.2 Partial Formalization of the Domain

Let $\mathcal{G}$ be the set of all RDF graphs, $\mathcal{O}$ be the set of all octet streams, and $\mathcal{M}$ be the set of all Internet media types. A representation is an octet stream typed by at least one Internet media type, but interaction protocols may define other ways to type those octet streams. We hence define an abstract set of *octet stream types* $\mathcal{T}$. Every octet stream type $t \in \mathcal{T}$ is at least associated with an Internet media type $\text{media}(t)$ that describes the Internet media type of the octet stream. Then, a *typed octet stream* is a pair $s = \langle o, t \rangle$ composed of an octet stream $o \in \mathcal{O}$ and a type $t \in \mathcal{T}$. The set of typed octet streams is written $\mathcal{S}$. We formalize RDF presentations as follows.

RDF presentation. An *RDF presentation* is a set of pairs $p \subseteq \mathcal{G} \times \mathcal{S}$ such that:

$$\forall \langle g, s \rangle, \langle g', s' \rangle \in p, \quad s = s' \Rightarrow g = g' \text{ and } \text{type}(s) = \text{type}(s') = t$$

The unique type t is the *type of the RDF representation*.

The set of first elements of p is the set of *valid* RDF graphs for p, and the set of second elements of p is that of *valid representations* of p. We use terms *RDF lifting*, and *RDF lowering* to refer to the process that consists of using an RDF presentation to decode a typed octet stream into an RDF graph, and the process that consists of using an RDF presentation to encode an RDF graph into a typed octet stream, respectively.[3] We formalize these terms as follows.

[3] Terms *lifting* and *lowering* were first mentioned in the W3C recommendation Semantic Annotation for WSDL and XML Schema - https://www.w3.org/TR/sawsdl/

Lifting Rule. $R^{\uparrow} : \mathcal{S} \rightarrow \mathcal{G}$ is a *lifting rule* for p if $(\forall g, s)[\langle g, s\rangle \in p \Rightarrow R^{\uparrow}(s) = g]$
Lowering Rule. $R^{\downarrow} : \mathcal{G} \rightarrow \mathcal{S}$ is a *lowering rule* for p if

$$(\forall g \in \mathcal{G}) \left[(\exists s \in \mathcal{S})[\langle g, s\rangle \in p] \Rightarrow (\exists s_0 \in \mathcal{S})[\langle g, s_0\rangle \in p \text{ and } R^{\downarrow}(g) = s_0]\right]$$

Valid RDF Graph. Let $p \in \mathcal{P}$ be a RDF presentation. A RDF graph g is *valid* for p iff $\exists s \in \mathcal{S}$ s.t. $\langle g, s\rangle \in p$. The *validation rule* of p is an application from $\mathcal{G}$ to $\{true, false\}$, that associates each graph $g \in \mathcal{G}$ with the value *true* if and only if g is valid for p.

Valid Representation. A typed octet stream s is a *valid representation* for p iff $\exists g \in \mathcal{G}$ s.t. $\langle g, s\rangle \in p$. The *representation validation rule* of p is the application of $\mathcal{S}$ to $\{true, false\}$, that associates each typed octet stream $s \in \mathcal{S}$ with the value *true* if and only if s is a valid representation for p.

We can propose some interpretation of existing RDF formats according to our definition of RDF presentation. For example, RDF/XML [2] is applicable to any RDF graph, and imposes that every RDF graph representation has the `application/rdf+xml` Internet media type. If we consider only Internet media types in $\mathcal{T}$, then RDF/XML defines a RDF presentation where every graph has an infinite number of representation (because of white spaces). From this RDF/XML presentations, one can define an infinite number of RDF sub-presentations that limit the set of valid RDF graphs. Any deterministic RDF/XML serializer defines a RDF presentation where every graph has exactly one representation, and whose domain is the set of all the RDF graph. The same interpretation can be made for Turtle [3]. For JSON-LD [28], that imposes RDF graph representations to have the Internet media type `application/json+ld`, there is no unique RDF presentation, because different JSON-LD contexts can make the same octet stream be lifted to different graphs. With one specific context however, one falls back to the same case as RDF/XML. RDF presentation is not a simple concept one would want to develop formalisms for, this is why we usually describe these in terms of their lifting rules, validation rules, lowering rules, or representation validation rules.

Many formalisms exist to specify lifting rules. We can cite the SPARQL-Generate rules [19,20], the RML mapping language [9], XSPARQL rules [1], the CSV metadata [29] and GRDDL [6].[4] Lowering rules can be specified using STTL [7] or XSPARQL [1]. Then, the SPIN formalisms [18], Shape expressions [22,12], and W3C SHACL, can be used to define validation rules for RDF graphs. As for formalisms that can be used to define representation validation rules, every XML-based Internet media types can be validated using XML Schema, while JSON Schema can validate those based on JSON.

In practice an extensive SPARQL-Generate lifting rule can be combined with a restrictive JSON-Schema representation validation rule to further specify what octet streams are valid (e.g., as a command for an actuator). Similarly, a STTL rule can be combined with a restrictive SHACL rule to further specify what graphs are valid (e.g., as the input of an OWL2 EL ontology documentation service).

[4] GRDDL defines an XML attribute used to link an XML files to a transformation (possibly XSLT) from XML to RDF/XML. If we consider only the transformations from XML files into valid RDF/XML files, then GRDDL transformation rules are lifting rules.

2.3 Proposal to extend the Linked Data principles

With the introduction of new interaction protocols for the Web (request/response protocols like the CoAP protocol for the Web of Things [27], or publish/subscribe protocols like WebSocket [10]), it becomes appropriate to propose to precise and extend the Linked Data principles [4]. In particular, we propose to include two additional rules that involve the concepts of RDF source and RDF presentation as defined above:

1. use IRIs to *identify* things;
2. use IRIs for which the interaction protocol defines a *look-up* mechanism (such as HTTP, CoAP, WebSocket);
3. when someone looks up a IRI, make so that they can retrieve a *description* of the resource identified by that IRI;
4. make so that the *representation* sent to the client is one of a RDF source;
5. make so that the client can find information about the RDF presentation of that representation, so that it can lift it back to RDF;
6. include links to other IRIs, so that they can discover more things.

In what follows, we assume that principles 1 to 4 are satisfied. The next sections are dedicated to the development of the RDF Presentation ontology, and Section 6 proposes practical solutions to satisfy principle 5 which is the key enabler for perfect understanding between web agents.

3 Perfect Understanding of Agents: Assumptions and Scenarios

In this section we propose four scenarios where agents communicate on the Web, all of which involve some adaptation of one or the other party to reach perfect understanding.

Let us first make the simple but useful assumption that the content to be shared is always that of an RDF graph. Even if the content is procedural, we assume some RDF graph can describe it using the appropriate vocabulary. Therefore, any message that is sent on the web is some representation of some RDF Graph. This allows us to adopt and work with the following paradigm: *Perfect understanding of agents is achieved when the RDF graph the sender encodes is equivalent to the RDF graph the receiver obtained after decoding the message*. It is important to note that this paradigm does not require every agent to implement RDF libraries, as RDF can just be some "abstract" model for the data it manipulates. Also, discussing entailment regimes is out of the scope of this work, but is planned in the future.

With the Web interaction protocols, the sender can be a client that sends a request to some server (e.g, using HTTP or CoAP), a server that answers to a client, or a publisher that broadcasts some message to its subscribers (e.g., using WebSocket or MQTT). Then, both agents can have any combination of the following features: (i) it may be constrained in some ways, i.e., it may have computational, storage, or battery constraints that prevents it from executing complex tasks; (ii) it may implement some of the principles we devise below, thus becoming what we call *semantically flexible*; (iii) it may request a trusted third party server to encode some content or decode some message and thus behave as if it was *semantically flexible*. Not all of the combinations enable

the agents to reach perfect understanding without prior hardwired agreement, but we will not discuss all of the possibilities. Instead, we discuss the following three main scenarios, which we detail in the rest of this section:

1. A server/publisher sends its message to a client/subscriber.
2. A client asks a server for the representation of a resource.
3. A client sends some encoded content to a server.

A server/publisher sends its message to a client/subscriber. Web agent 1 sends an RDF graph encoded using a presentation that is unknown to web agent 2. Web agent 2 may have issued an HTTP GET to some URI of web agent 1, or web agent 2 may be observing web agent 1. Web agent 2 cannot decode the message properly, unless it discovers how this message can be lifted to RDF. If web agent 1 is an HTTP server it could include such metadata in its response header, or the web agent 2 could also discover that metadata elsewhere on the Web, such as in a document that describes the server.

If web agent 1 is constrained, it can therefore send messages in lightweight and application-specific formats. If it doesn't send information about how to lift the message, web agent 2 can still discover this information somewhere else. If now web agent 2 itself is constrained, it can rely on a trusted translation server to lift the message for him.

A client asks a server for the representation of a resource. A web client issues an HTTP GET at some IRI, and would like the answer to be encoded following a specific RDF presentation so that it can lift it. The server cannot lower a response the client will be able to lift, unless it discovers somehow how to lower the response RDF graph properly. The client could include such metadata in the request header.

If the client is constrained, it can thus request messages in a format that is optimized to its use or its constraints. If the server is constrained but is aware of our approach, it can rely on a trusted translation server to generate a response so that the client can access to its meaning. This could include redirecting the client to the translation server.

A client sends some encoded content to a server. Suppose a web server only accepts requests presented in some ways, which is unknown by a potential client. The client cannot lower its request RDF graph, unless it discovers some information about the presentation the server uses. The server could include such metadata in the header of every response, or the client could discover that metadata elsewhere on the Web, such as in a document that describes the server.

If the server is constrained, it can then request messages in formats that are optimized for its use. If now the client itself is constrained, it can rely on a trusted translation server to lift the request for him.

4 Competency Questions

Based on these three scenarios, we extract a list of competency questions that must be answered in this paper. These competency questions are not meant to be solely answered

by the RDF Presentation ontology we will develop in Section 5. In fact, CQ5-8 are answered using RDF Presentation discovery techniques we will devise in Section 6. The last three competency questions involve concepts from the SOSA/SSN ontology and the WoT TD ontology.

CQ1: What are the RDF lifting and lowering procedures one can use for a specific RDF presentation?
CQ2: What is type of the octet streams a RDF presentation lowers RDF graphs into?
CQ3: Is a RDF graph valid for a certain RDF presentation?
CQ4: Is an octet stream valid for a certain RDF presentation?
CQ5: How can a server inform its client on the RDF presentation it can use to lift the message?
CQ6: How can a server directly inform its client of the RDF lifting procedure it can use to lift the message?
CQ7: How can a client negotiate the use of a specific RDF presentation with its server?
CQ8: How can a client negotiate the use of a specific RDF lowering procedure with its server?
CQ9: How can one describe the RDF presentation that is used in the input of an actuator?
CQ10: How can one describe the RDF presentation that is used in the output of a sensor?
CQ11: How can one describe the RDF presentation that is used in the input and output of the interaction pattern a thing implements?

5 RDFP Ontology and Alignments to SOSA/SSN and WoT TD

This section introduces the RDF Presentation ontology (RDFP), built to model RDF presentations, and their associated lifting, lowering, validation, procedures. We introduce the RDFP ontology itself, then propose alignments to SOSA/SSN and WoT TD with which it has practical use.

The RDFP Ontology. RDFP defines terms in namespace `https://w3id.org/rdfp/` that we shorten rdfp:. It is also identified by rdfp:, and is documented and published there following the Linked Open Vocabulary best practices.

The RDFP Ontology defines class rdfp:GraphDescription to describe RDF Graphs in terms of presentation and representations. A rdfp:GraphDescription may be linked to some rdfp:GraphRepresentation using rdfp:representedBy, to some rdfp:GraphPresentation using rdfp:presentedBy, and to rdfp:ValidationRule using rdfp:validationRule. Class rdfp:GraphPresentation describes RDF Presentations. They may be link to (1) their type using property rdfp:mediaType, (2) some rdfp:GraphValidationRule using also rdfp:validatedBy, (3) some rdfp:LiftingRule using also rdfp:liftingRule, and (4) some rdfp:LoweringRule using rdfp:loweringRule. We plan to rename relations in the future to make them differentiable from class names otherwise than with just case distinction. As an example of use, the following Turtle snippet describes an actual RDF presentation that is used on the demonstration website of RDFP, with URI `https://w3id.org/rdfp/example/graph/xml`.

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

<graph/xml> a rdfp:GraphPresentation ; // an RDF presentation
  rdfp:mediaType "application/xml" ;
  rdfp:liftingRule <graph/xml/liftingRule> ;
  rdfp:loweringRule <graph/xml/loweringRule> ;
  rdfp:presentationFor <graph> .
<graph> a rdfp:GraphDescription ; // an RDF graph description
  rdfp:presentatedBy <graph/xml> , <graph/json> ;
  rdfp:validationRule <validationRule> .
<sensorOutput154> a rdfp:Graph ; rdfp:describedBy <graph> . // a specific RDF graph
```

This RDF presentation has Internet media type `application/xml`, and is partially defined in terms of lowering, lifting, and validation rules. These rules are identified by IRIs, and their representation can be obtained at these IRIs. For example if one looks up the IRI of lifting rule `https://w3id.org/rdfp/example/graph/xml/liftingRule`, one retrieves a SPARQL-Generate query. Other descriptions of that rule could also be retrievable at that same IRI, such as a RML mapping, or a XSPARQL query.

Alone, the RDFP ontology answers Competency Questions CQ1-4.

Alignment to SOSA/SSN. The input and output of a sosa:Procedure can be described using RDF Presentations (i.e., using instances of rdfp:GraphPresentation), with their associated lifting, lowering, and validation rules. Then, a sosa:Result may be an instance of rdfp:GraphDescription that would be presented using that RDF Presentation. Example [15, §B.9] in the SOSA/SSN specification actually uses RDFP to model how the output of a DHT22 temperature and humidity sensor can be validated against some SHACL shapes. This answers Competency Questions CQ9 and CQ10.

Alignment to WoT TD. We already proposed some alignment between SOSA/SSN and TD in the past,[5] which has been discussed by the WoT group. A td:Thing can be considered as a ssn:System which implements interaction patterns (sosa:Procedures) with some input and output description. TD additionally models the access point where one may trigger executions of these procedures using class td:Link. This answers Competency Question CQ11.

```
td:Thing rdfs:subClassOf ssn:System .
td:interaction rdfs:subPropertyOf ssn:implements .
td:InteractionPattern rdfs:subClassOf sosa:Procedure .
td:inputData rdfs:subPropertyOf ssn:hasInput .
td:outputData rdfs:subPropertyOf ssn:hasOutput .
```

The class td:DataSchema in WoT TD can therefore be aligned to td:GraphPresentation, thus allowing expressive and flexible description of the input and output of interaction patterns of a td:Thing. Using RDFP in WoT TD could void the need for developing yet another type system in various data formats (XML, JSON).

Next section proposes practical solutions to enable the scenarios of Section 3, satisfy principle 5 of our extended Linked Data principles, and answer Competency Questions CQ5-8.

[5] https://lists.w3.org/Archives/Public/public-wot-ig/2017Jul/0008.html

6 Practical solutions for Perfect Understanding on the Web

In this section we propose means to enable the scenarios of Section 3 on two Web interaction protocols that define a *look-up* mechanism: HTTP and CoAP. Section 6.1 first focuses on the direct discovery of information about the RDF Presentation, i.e., scenarios where some information is directly included in the request or the response. Then, Section 6.2 overviews practical solutions for the other cases.

6.1 Direct discovery of information about the RDF presentations

What can lead to the realization of the scenarios from Section 3 is *bindings* onto the interaction protocols of the Web. For the client and the web server, they have to have a way to discover information about the used RDF presentation or the one they have to use. We limit here the frame of our study in two ways:

- we consider only the direct discovery of information about the RDF Presentation scenarios, that is to say scenarios where part of the information is directly included in the request or the response;
- We consider only the HTTP and CoAP protocols. However, similar implementations could be proposed for other interaction protocols as well.

The *Constrained Application Protocol (CoAP)* [26] is a Web transfer protocol designed to address requirements specific to constrained devices. For instance, while HTTP is a text-based protocol, CoAP is a binary protocol that emphasizes message conciseness. However, CoAP integrates seamlessly with HTTP: HTTP methods, response codes, media types and some header fields have a one-to-one mapping to CoAP equivalents. Therefore, in what follows, we focus our discussion on the more widely known HTTP protocol, but the *HTTP header fields* we introduce in this section can also be easily defined as equivalent *CoAP options* [26].

RDF presentations used. The RDF presentation qualifies the link between the RDF source and the way it is represented. As we follow the Web's architectural principles seen in the section 2.1, we want to keep orthogonal the identification and presentation concepts. We claim that the RDF presentation is part of what types the representation octet stream, and not the resource itself. It would be convenient to link the RDF presentation to its Internet media type used using an Internet media type parameter like for example: `application/seas;p="https://w3id.org/rdfp/example/graph"`. However, even if that parameter can be defined for new Internet media type, it is not possible to define a global parameter for every Internet media type, as imposed by RFC 2045 [11]:

> 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.

Hence, we introduce a new HTTP header field `Content-Presentation` for that purpose. The value of that field is an absolute IRI that identifies an RDF presentation.

By adding such header fields to 200 OK responses, an existing service can be adapted at lower costs to the formalisms of the Semantic Web and can do "as if" it was exposing RDF sources. The client would just have to follow these steps: (i) look up for the RDF presentation IRI, (ii) parse the retrieved RDF graph in order to find the associated lifting rule IRI, (iii) look up that IRI, and (iv) use some representation of that rule to interpret the message in RDF. That approach can be used for constrained servers as well, allowing them to send messages in format as light and specific to their application as needed. As long as the clients can discover the RDF presentation used, they can interpret the message in RDF. If the client itself is constrained, he can also rely on a trusted third-party server to do the transformation.

Negotiating the RDF presentation to be used. With the RDF presentation negotiation, the client can specify its preferences about the RDF presentation that has to be used to represent the RDF graph in the body of the HTTP response.

One solution to allow the client to transmit that information to the server is to introduce an HTTP header field `Accept-Presentation`. The value of that field is an absolute IRI that identifies the RDF presentation that the client wishes to see used by the server.

By using that header, the constrained client can ask a server to bring the answer in a specific format. If the server itself is constrained, it can ask another trusted third-party server to do the transformation.

Pointing directly to a rule. In some situations, it seems unreasonable to expect that a client/server will: (i) look up for the RDF presentation IRI, (ii) parse the retrieved RDF graph in order to find the associated lifting rule IRI, (iii) look up that IRI, and (iv) use some representation of that rule to interpret the message in RDF.

In the simple cases, which also represent most of the cases, knowing one of those rules is enough. We hence propose to introduce additional HTTP header fields to point directly to those rules. For instance:

- the HTTP header `Content-Lifting-Rule` means that the web agent that receives the representation can use the lifting rule identified by that IRI to lift the representation to the RDF graph. If the representation is that of a resource, then that RDF Graph is isomorphic to the content of that resource (a RDF source);
- the HTTP header `Accept-Lowering-Rule` means that the client asks the server to use the lowering rule identified by that IRI to generate a RDF graph. That RDF graph is then a description of the resource identified by the IRI the client looked up.

6.2 Indirect discovery of information about the RDF presentations

Indirect discovery covers cases where a web client discovers some information about a RDF presentation prior to accessing a RDF Source that uses this RDF presentation. This information could be in RDF or in another format such as the Web Links [21]. It could be part of the content of any RDF Source, some *thing description* using the WoT TD ontology on the Web of Things[6], or a CoRE Link Format [25] document.

[6] The description of things and the discovery of those descriptions is one of the focus of the W3C interest group for the Web of Things

Description in RDF. Apart from describing RDF presentations and their different types of rules, the RDFP ontology defines the class `rdfp:WebDocument` of web documents, and its sub-class `rdfp:RDFSource` of RDF sources. It is expected that the IRI of a `rdfp:WebDocument` uses an interaction protocol that defines a look-up mechanism. An RDF source can be linked to a unique graph description using property `rdfp:describedBy`, which can then be linked to zero or more RDF Presentations. For example, the following Turtle snippet asserts that resource identified by IRI `http://ci.emse.fr/rdfp/example/input` is a RDF source, and is described by `https://w3id.org/rdfp/example/graph`.

```
<http://ci.emse.fr/rdfp/example/input> a rdfp:RDFSource ;
  rdfp:describedBy <graph> .
```

RDF Graphs that use the RDFP ontology can hence describe RDF sources and the way they can be presented. The publisher of some RDF source can also advertise about the available RDF presentations for this RDF source by publishing such graphs elsewhere on the server or on the Web. User agents can thus discover the RDF presentation of the RDF source.

Description as Web Links. CoAP servers typically expose the resources they host in the CoRE Link Format [25]. The CoRE Link Format is a format for documents that contain only Web Links [21]. [25] defines the *hosts* Web Linking relation, which denotes a relations from a server to a hosted resource, and is the default in CoRE Link Format documents. New link relations could be defined to enable the indirect discovery of information about a RDF presentation:

```
<http://ex.org/lower>;rel="lowering";anchor="/actuator",
<http://ex.org/lift>;rel="lifting";anchor="/sensor"
```

A second option to reach the same goal would be to rely on new link attributes instead. More work is needed to decide which of these two options is preferable.

```
</actuator>;rt="light-switch";if="POST";lowering="http://ex.org/lower",
</sensor>;rt="presence-sensor";if="GET";lifting="http://ex.org/lift"
```

7 Conclusion

In this paper, we paved the way for a Web where things and services can discover how they can reach perfect understanding of the data they exchange. We first extended and formalized some definitions of the W3C documents that build the foundations of the Web and the Web of Data. That formalization helped us to define the notion of RDF presentation, but also the concepts of validation, lifting, lowering and representation validation rules. Each of these rules can be expressed using different existing formalisms. On top of these notion, we proposed an extension of the Linked Data principles.

We then proposed practical solutions to enable the scenarios of section 3 using existing formalisms and standards. Our RDF Presentation ontology can be used to describe information about RDF sources and RDF presentations on the Web. We showed how it can be used in combination with the OGC and W3C joint SOSA/SSN standard ontology. We also explained how it could drastically empower the W3C WoT Thing Description to describe the input and output dataSchema of thing interaction patterns.

We showed that similar description as with the RDFP ontology could be obtained using the CoRE Link Format for constrained devices. Solutions to directly discover such information have been proposed for the HTTP and the CoAP interaction protocols.

RDFP has been exemplified in a section of the SOSA/SSN ontology specification [15, §B.9], which should help to encourage its use. In fact, all it takes to adapt some service or thing already in place to RDFP and our extended Linked Data principles is (i) either to add a single HTTP header with the URL of a RDF Presentation, or (ii) expose some RDFP description somewhere that describes the existing service or thing, potentially using TD. This should be more affordable than adding support for some new format while maintaining the legacy consumers of the service or device.

In the future we plan to work on checking formally that for example a SPARQL-Generate rule is indeed symmetric to a STTL rule, precise the proposed solutions for CoAP and using the CoRE Link Format, and to showcase these on a set of constrained things that use a variety of formats. Combining this with legacy servers, resource discovery servers, and translation servers, this would form a global ecosystem of objects and services that use different formats but remain interoperable semantically.

Acknowledgments. This work has been partly funded by the ANR 14-CE24-0029 OpenSensingCity project, and a bilateral research convention between ARMINES Fayol and ENGIE R&D.

References

1. Waseem Akhtar, Jacek Kopeckỳ, Thomas Krennwallner, and Axel Polleres. XSPARQL: Traveling between the XML and RDF worlds–and avoiding the XSLT pilgrimage. In *European Semantic Web Conference*, pages 432–447. Springer, 2008.
2. David Beckett. RDF/XML Syntax Specification (Revised) W3C Recommendation 10 February 2004. W3C Recommendation, W3C, February 10 2004.
3. David Beckett and Tim Berners-Lee. Turtle - Terse RDF Triple Language, W3C Team Submission 14 January 2008. W3C team submission, W3C, January 14 2008.
4. Tim Berners-Lee. Linked data. Published online at http://www.w3.org/DesignIssues/LinkedData.html, 2005. W3C Design issue.
5. Carsten Bormann and Hoffman Paul. Concise Binary Object Representation (CBOR). IETF RFC 7049, October 2014.
6. Dan Connolly. Gleaning Resource Descriptions from Dialects of Languages (GRDDL), W3C Recommendation 11 September 2007. W3C Recommendation, W3C, September 11 2007.
7. Olivier Corby and Catherine Faron-Zucker. Sttl: A sparql-based transformation language for rdf. In *11th International Conference on Web Information Systems and Technologies*, 2015.
8. Richard Cyganiak, David Wood, and Markus Lanthaler. RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation 25 February 2014. W3C Recommendation, W3C, February 25 2014.
9. Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. Rml: A generic language for integrated rdf mappings of heterogeneous data. In *LDOW*, 2014.
10. I. Fette and A. Melnikov. The WebSocket Protocol. RFC 6455 (Proposed Standard), December 2011.

11. Ned Freed and Borenstein Nathaniel. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. IETF RFC 2045, November 1996.
12. Jose Emilio Labra Gayo, Eric Prud'hommeaux, Harold R Solbrig, and Jose María Álvarez Rodríguez. Validating and Describing Linked Data Portals using RDF Shape Expressions. In *Workshop on Linked Data Quality, SEMANTICS*. Citeseer, 2014.
13. Dominique Guinard and Vlad Trifa. 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, April 2009.
14. Dominique Guinard, Vlad Trifa, and Erik Wilde. A resource oriented architecture for the web of things. In *Internet of Things (IOT), 2010*, pages 1–8. IEEE, 2010.
15. Armin Haller, Krzysztof Janowicz, Simon J D Cox, Danh Le Phuoc, Kerry Taylor, and Maxime Lefrançois. Semantic Sensor Network Ontology. W3C Recommendation, W3C, October 19 2017.
16. Ian Jacobs and Norman Walsh. Architecture of the World Wide Web, Volume One, W3C Recommendation 15 December 2004. W3C Recommendation, W3C, December 15 2004.
17. Sebastian Kaebisch and Takuki Kamiya. Web of Things (WoT) Thing Description. First Public Working Draft, W3C, September 14 2017.
18. Holger Knublauch. SPIN-modeling vocabulary. *W3C Member Submission*, 22, 2011.
19. Maxime Lefrançois, Antoine Zimmermann, and Noorani Bakerally. 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, November 2016.
20. Maxime Lefrançois, Antoine Zimmermann, and Noorani Bakerally. 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, January 2017.
21. M. Nottingham. Web Linking. RFC 5988 (Proposed Standard), October 2010.
22. Eric Prud'hommeaux, Jose Emilio Labra Gayo, and Harold Solbrig. Shape expressions: an RDF validation and transformation language. In *Proceedings of the 10th International Conference on Semantic Systems*, pages 32–40. ACM, 2014.
23. Leo Sauermann and Richard Cyganiak. Cool URIs for the Semantic Web. W3C Note, W3C, December 03 2008.
24. John Schneider, Takuki Kamiya, Daniel Peinter, and Rumen Kyusakov. Efficient XML Interchange (EXI) Format 1.0 (Second Edition). W3C Recommendation, W3C, February 11 2014.
25. Z. Shelby. Constrained RESTful Environments (CoRE) Link Format. RFC 6690 (Proposed Standard), August 2012.
26. Z. Shelby, K. Hartke, and C. Bormann. The Constrained Application Protocol (CoAP). RFC 7252 (Proposed Standard), June 2014.
27. Zach Shelby, Klaus Hartke, and Carsten Bormann. The constrained application protocol (CoAP). IETF RFC 7252, October 2014.
28. Manu Sporny, Gregg Kellogg, and Markus Lanthaler. A JSON-based Serialization for Linked Data. W3C Recommendation, W3C, January 16 2014.
29. Jeni Tennison and Gregg Kellogg. Model for Tabular Data and Metadata on the Web. W3C Recommendation, W3C, December 17 2015.
30. Erik Wilde. Putting things to REST. *School of Information*, 2007.