The Resource Description Framework (RDF)

Overview

Although it was originally designed as a metadata data model, *Resource Description Framework* (RDF) has become a general web resource description and modeling language. It can be used for conceptual description or modeling of information stored in web resources. RDF can be used to create a machine-readable description about any kind of resource, because RDF files can be extended with an arbitrary number of external vocabularies. In contrast to many W3C standards, RDF has no single specification but is defined by a set of documents [1].

The RDF data model can be used for describing any kind of resources that can be identified by a URI. As mentioned earlier, an RDF document is a sequence of statements called *RDF triples* (resource–property–value or subject–predicate–object). The predicate (property) that denotes a relationship between the subject and the object can be binary only. Any expression in RDF is a collection of triples. A set of triples is called an RDF graph, which is a directed, labeled graph that represents information on the Web. The nodes of the RDF graph are the resources and values [2].

As shown earlier, a person can be described using the FOAF vocabulary. Such descriptions can be written either in XML or in RDF. Listing 1 shows how to write FOAF in RDF.

Listing 1. Describing a Person in RDF

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:foaf="http://xmlns.com/foaf/0.1/"
 xmlns="http://www.example.com/johnsmith/contact.rdf#">
  <foaf:Person rdf:about="http://www.example.com/johnsmith/contact.rdf#johnsmith">
    <foaf:mbox rdf:resource="mailto:john.smith@example.com" />
    <foaf:homepage rdf:resource="http://www.example.com/johnsmith/" />
    <foaf:family_name>Smith</foaf:family_name>
    <foaf:givenname>John</foaf:givenname>
  </foaf:Person>
</rdf:RDF>
```

Figure 1 represents this file as an RDF graph.

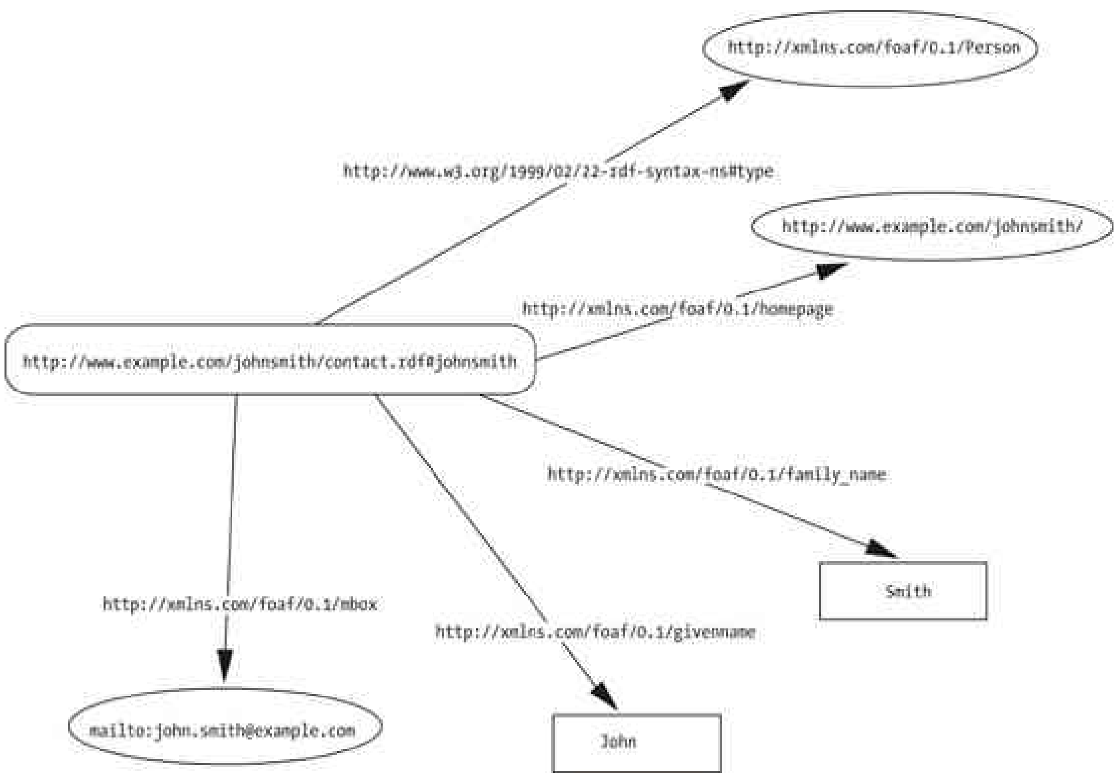
http://xmlns.com/foaf/0.1/Person
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.example.com/johnsmith/
http://xmlns.com/foaf/0.1/homepage
http://www.example.com/johnsmith/contact.rdf#johnsmith
http://xmlns.com/foaf/0.1/family_name
Smith
http://xmlns.com/foaf/0.1/mbox
http://xmlns.com/foaf/0.1/givenname
mailto:john.smith@example.com
John

Figure 1. A simple RDF graph

As you will see, RDF can be expressed in a variety of formats. For example, Listing 2 is another notation of Listing 1. This notation, N3, will also be described later in detail.

Listing 2. The N3 Equivalent of the Previous Example

```
@prefix :     <http://www.example.org/~joe/contact.rdf#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

:joesmith a foaf:Person ;
  foaf:givenname "Joe" ;
  foaf:family_name "Smith" ;
  foaf:homepage <http://www.example.org/~joe/> ;
  foaf:mbox <mailto:joe.smith@example.org> .
```

The RDF namespace is `http://www.w3.org/1999/02/22-rdf-syntax-ns#`, which is conventionally associated with the namespace prefix `rdf:`.

The Unicode strings in URI references of RDF graphs cannot contain control characters (`#x00`–`#x1F`, `#x7F`–`#x9F`). These URIs should be absolute URIs with optional fragment identifiers.

RDF literals are used to identify values such as numbers and dates. RDF literals are Unicode strings containing one or two named components. They should be written in *UTF-8 normalized in Normalization Form C* (Canonical Decomposition followed by Canonical Composition [3]). RDF literals can be either plain or typed. Plain literals are strings combined with an optional language tag (normalized to lowercase). They correspond to plain text in a natural language. Typed literals are strings combined with a datatype URI for applying the lexical-to-value mapping to the literal string.

The Formal Grammar of the Resource Description Framework was introduced in 1999 [4]. RDF has the following vocabulary:

- `rdf:Alt`, `rdf:Bag`, `rdf:Seq`
  Containers of alternatives, unordered containers, and ordered containers (rdfs:Container is a superclass of the three)
- `rdf:List`
  The class of RDF lists
- `rdf:nil`
  An empty list (an instance of rdf:List)
- `rdf:Property`
  The class of properties
- `rdf:Statement`, `rdf:subject`, `rdf:predicate`, `rdf:object`
  Reification
- `rdf:type`
  A predicate that identifies the class that the resource is an instance of
- `rdf:XMLLiteral`
  The class of typed literals

This vocabulary is also used as the basis for the extensible knowledge representation language *RDF Schema*.

There is a *query language* called *SPARQL* (pronounced “Sparkle”) that can be used to retrieve and manipulate information stored in RDF or in any format that can be retrieved as RDF [5]. The output can be a results set or an RDF graph. It is also possible to update RDF graphs through a protocol known as the *SPARQL 1.1 Uniform HTTP Protocol* [6].

The Resource Description Framework technology is important from the standardization point of view for many reasons. First, the basic data model of RDF is a standard graph. Second, the naming system applies standard URLs. The data retrieval and composition mechanisms used by RDF are also standard technologies.

RDF can be provided in a variety of syntaxes/serialization formats, for example, RDF XML serialization (RDF/XML), RDFa, Turtle, Notation3, JSON-LD, N Triples [7], TRiG [8], and TRiX [9]. The most common ones are described in the next sections.

Major RDF syntaxes

RDF in XML Serialization Syntax

The recommended and most frequently used syntax for RDF applications is the XML serialization format, RDF/XML [10]. Although there are other notations of RDF that are easier to read and write (see the next sections), RDF/XML provides widely accepted XML documents. However, the fundamental problem with RDF/XML is the contradiction of representing a graph with a tree structure.

The Internet media type for RDF/XML is `application/rdf+xml`. The recommended file extension is `.rdf`.

The XML serialization of RDF provides well-formed XML documents.

A person’s Wikipedia page, for example, can be described in a machine-readable form in RDF/XML, as shown in Listing 3.

Listing 3. A Wikipedia Page Described in RDF/XML

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://en.wikipedia.org/wiki/Rowan_Atkinson">
    <dc:title>Rowan Atkinson</dc:title>
    <dc:publisher>Wikipedia</dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

RDF in N3 Syntax

Notation 3, often abbreviated as *N3*, is a shorthand non-XML serialization of RDF. It is a superset of RDF and is more compact than the XML serialization of RDF. The grammar of N3 is defined by W3C in many formats [11], for example, in *Extended Backus-Naur Form* (EBNF) [12].

The MIME type and character encoding of N3 should be declared as `text/n3; charset=utf-8`. The typical file extension is `.n3`. Tokenizing and whitespace handling are not specified in the grammar.

Base URIs to be used for the parsing of relative URIs can be set with the `@base` directive in the form `@base <http://example.com/overview/>`.

A prefix can be associated to a namespace URI by the `@prefix` directive.

Several rules for string escaping are derived from Python, namely, `stringliteral`, `stringprefix`, `shortstring`, `shortstringitem`, `longstring`, `longstringitem`, `shortstringchar`, and `longstringchar`. Additionally, the `\U` extension, also used in another RDF serialization (N-Triples), can be applied. Legal escape sequences are `\newline`, `\\` (backslash, `\`), `\'` (single quote, `'`), `\"` (double quote, `"`), `\n` (ASCII Linefeed, LF), `\r` (ASCII Carriage Return, CR), `\t` (ASCII Horizontal Tab, TAB), `\uhhhh` (Unicode character in BMP), and `\U00hhhhhh` (Unicode character in plane 1–16 notation). The escapes `\a`, `\b`, `\f`, and `\v` cannot be used because the corresponding characters are not allowed in RDF.

Shorthand notation can be used for the following common predicates:

- `a` (stands for `<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>`)
- `=` (stand for `<http://www.w3.org/2002/07/owl#sameAs>`)
- `=>` (stands for `<http://www.w3.org/2000/10/swap/log#implies>`)
- `<=` (stands for `<http://www.w3.org/2000/10/swap/log#implies>`)

New classes and new properties can be defined in new vocabularies [13]. A class can be defined as Listing 4 because the `rdf:type` property is abbreviated as a in N3.

Listing 4. An RDF Class in N3

```
:Sport a rdfs:Class.
```

An object of the class can be defined as shown in Listing 5.

Listing 5. Declare an Object of a Class in N3

```
:Kayak a :Sport.
```

Objects can be in multiple classes. Relationships between classes can be written as shown in Listing 6.

Listing 6. Class Relationships in N3

```
:Watersport a rdfs:Class; rdfs:subClassOf :Sport .
```

A property can be defined as shown in Listing 7.

Listing 7. Property Declaration in N3

```
:paddle a rdf:Property.
```

Relationships between classes are not necessarily hierarchical relationships. You can see an example in Listing 8.

Listing 8. Nonhierarchical Relationships in N3

```
:paddle rdfs:domain :Sport;
rdfs:range :Watersport.
```

The person’s Wikipedia page described in the previous section can be written in N3, as shown in Listing 9.

Listing 9. A Wikipedia Page Description in N3

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.

<http://en.wikipedia.org/wiki/Rowan_Atkinson>
  dc:title "Rowan Atkinson";
  dc:publisher "Wikipedia".
```

Notation3 has several subsets, including Turtle, N-Triples, N3 RDF, and N3 Rules. The most popular of them is discussed in the next section.

RDF in Turtle Syntax

A subset of N3 is the *Terse RDF Triple Language*, often referred to as Turtle. Turtle provides a syntax to describe RDF graphs in a compact textual form, which is easy to develop. It is a subset of Notation 3 (N3) and a superset of N-Triples. Turtle is popular among Semantic Web developers and considered as an easy-to-read alternative to RDF/XML. Turtle is being standardized by the World Wide Web Consortium [14]. The typical file extension of Turtle files is `.ttl`. The character encoding of Turtle files should be UTF-8. The MIME type of Turtle is `text/turtle`. Turtle is supported by many software frameworks that can be used for querying and analyzing RDF data, such as Jena [15], Redland [16], and Sesame [17].

Turtle files consist of a sequence of directives, statements representing triples, and blank lines. Triples can be written in Turtle as a sequence of subject – predicate – object terms, separated by whitespace, and terminated by a period (`.`). URIs should be written in angle brackets (`<>`). Literals are delimited by double quotes (`""`). Listing 10 shows an example.

Listing 10. A Basic Example for the Turtle Syntax

```
<http://example.com/shop> <http://example.com/contact> "Text content"
.
```

URI length can be reduced by the `@PREFIX` (Listing 11).

Listing 11. A URI Prefix Declaration

```
@PREFIX ex: <http://example.com/> .
```

In that case, the first example can be written as in Listing 12.

Listing 12. Using a Prefix

```
ex:shop ex:contact "Text content" .
```

where `ex:shop` declares the concatenation of `http://example.com/` with `shop`, revealing the original URI `http://example.com/shop`.

RDFa

The power of RDF, which was demonstrated earlier, can be exploited through external files written in rather complex syntax. However, there is a nice exception: RDFa. RDFa (RDF in attributes) adds attribute-level extensions to any markup language (from this point of view, the host language) in order to describe structured data. In other words, RDFa notations can be declared in attributes, rather than elements (which is the approach used by other RDF serialization formats). Although many attributes are defined by RDFa, some markup attributes (such as `href` and `rel`) are reused. Wherever possible, the textual content is also reused. RDFa can serve as a bridge between the “human and data Webs,” since RDFa makes it possible to write RDF triples in the (X)HTML markup [18]. Structured information can be extracted and utilized from web documents via an RDFa application programming interface (RDFa API) [19]. The mechanism of the RDF data model mapping allows RDF triples to be embedded within web documents as well as the extraction of RDF model triples by compliant software.

RDFa provides the option to embed rich metadata within certain attributes of web documents [20]. The set of attributes to be used for this purpose is as follows:

- `about`, `src`
  The Unified Resource Identifier (URI) or compact URI (CURIE) [21] of the resource that describes the metadata
- `rel`, `rev`
  Relationship with another resource
- `href`, `resource`
  The partner resource
- `property`
  A property for the content of an element
- `content`
  Element content override when using the property attribute (optional)
- `datatype`
  The datatype of text specified for use with the property attribute (optional)
- `typeof`
  The RDF type(s) of the subject (optional)

RDFa makes it possible to arbitrarily mix multiple independently developed vocabularies. It can be parsed without analyzing the specific vocabularies being applied. This is one of the most advanced technologies to provide different types of machine-readable structured data in the markup.

Since the “a” in RDFa stands for attributes whose styles are provided most commonly in Cascading Style Sheets, it is straightforward to use CSS selectors to style the code [22]. For example, if the name of the creator and the book title of the previous example appear throughout the site, all instances can be styled using universal selectors (Listing 13).

Listing 13. Styling RDFa

```
* [property="dc:creator"]  {
  color: #2a56d3;
  font-style: italic;
}
* [property="dc:title"] {
  font-size: 2em;
  font-family: "Georgia";
}
```

The latest news on RDFa can be tracked on the web site of the RDFa Working Group of W3C at `www.w3.org/2010/02/rdfa/` [23].

For example, let’s describe a person with RDFa notation using the FOAF vocabulary! First we need to declare the FOAF namespace (either in the document head or on the body element). The about attribute of RDFa can be used to express the subject, while the RDFa attribute property sets the predicate (Listing 14).

Listing 7-14. An RDFa Annotation Using FOAF

```
<body xmlns:foaf="http://xlmns.com/foaf/0.1/">
  <p about="#smith" property="foaf:name">John Smith</p>
</body>
```

The content of the `p` element is both a human- and machine-readable text that will be rendered on the web page.

Now extend the previous example with another person and express a relationship between the two persons (Listing 7-15)! The class of the entity can be declared by the `typeof` attribute. In this case, we use the `Person` class from the FOAF vocabulary to “let the browser know” that John Smith is a person. The second person is declared exactly the same way. Finally, we use the term `knows` from the FOAF vocabulary and pass it as the value of the `rel` attribute to express that John Smith knows Peter Johnson (declared by the `resource` attribute).

Listing 15. Two People and the Relationship Between Them Expressed Using FOAF in RDFa

```
<body xmlns:foaf="http://xmlns.com/foaf/0.1">
  <p>
    <span about="#john" typeof="foaf:Person" property="foaf:name">John Smith</span>
is
    interested in smartphones. <span about="#jane" typeof="foaf:Person"
    property="foaf:name">Peter Johnson</span> is an Android developer. <span
    about="#john" rel="foaf:knows" resource="#peter">John and Peter knows each
```

```
other.
    </span>
  </p>
</body>
```

Compare this machine-readable statement with MySQL database records displayed using PHP, and you have a glimpse of the power of the Semantic Web!

Other vocabularies can be similarly used with RDFa. For example, Dublin Core metadata can be embedded to the markup using RDFa, as shown in Listing 16.

Listing 16. An RDFa Annotation Using DC

```
<p xmlns:dc="http://purl.org/dc/elements/1.1/"
 about="#standardweb" property="dc:title">
 Web standardista <span about="#sikos" property="dc:creator">Dr. Sikos</span>
describes
 Web standardization, accessibility, and Web semantics in his latest book
 <cite about="#webstandards" property="dc:title">Web standards</cite>. The first
press
 release has been published on <span about="#webstandards" property="dc:date"
 content="2011-11-14">14 November 2011</span>.
</p>
```

RDF Schema

According to the W3C Metadata Activity, RDF Schema (RDFS) is “a declarative representation language influenced by ideas from knowledge representation” [24]. RDF Schema extends RDF with structure (classes, properties of properties, etc.). It can be used to formalize metadata exchange between human-readable and machine-processable vocabularies. Beyond the basic RDF vocabulary discussed earlier, RDFS has several additional constructs [25]:

- Classes
    - `rdf:Property`
    - `rdf:XMLLiteral`
    - `rdfs:Class`
    - `rdfs:Datatype`
    - `rdfs:Literal`
    - `rdfs:Resource`
- Properties
    - `rdf:type`
    - `rdfs:comment`
    - `rdfs:domain`
    - `rdfs:isDefinedBy`
    - `rdfs:label`
    - `rdfs:range`

- `rdfs:seeAlso`
- `rdfs:subClassOf`
- `rdfs:subPropertyOf`

These classes and properties provide a more advanced level of knowledge representation than RDF does and can be used for basic description of web ontologies. This is the reason why the more expressive language Web Ontology Language (OWL) reuses many RDFS components.

For example, the resource “macaw” can be declared as a subclass of the class “birds,” as shown in Listing 7-69.

Listing 7-69. A Simple RDFS Example

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xml:base="http://www.example.com/birds#">
  <rdf:Description rdf:ID="bird">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  </rdf:Description>
  <rdf:Description rdf:ID="macaw">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#bird"/>
  </rdf:Description>
</rdf:RDF>
```

This notation can also be shortened by using `rdfs:Class` instead of `rdf:Description` and omitting `rdf:type` (Listing 7-70).

Listing 7-70. An Optimized Version of the Previous Example

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xml:base="http://www.example.com/birds#">
  <rdfs:Class rdf:ID="bird" />
    <rdfs:Class rdf:ID="macaw">
    <rdfs:subClassOf rdf:resource="#bird"/>
  </rdfs:Class>
</rdf:RDF>
```

See also