

Как представлять библиографические данные в RDF

Говоря о представлении библиографических данных в RDF, мы должны всегда помнить, для чего мы это делаем. Цель, которую мы преследуем – сделать библиографическую информацию, накопленную библиотеками, открытой и доступной для стандартных поисковых машин и web-браузеров; используя методологию Связанных данных, т.е. путем публикации структурированных данных и установления связей между данными из разных источников с соблюдением следующих принципов:

- все описываемые сущности получают уникальные идентификаторы – URI
- для публикации данных используется модель данных RDF;
- данные из разных источников связываются с помощью RDF-связей

Основы RDF

RDF (Resource Description Framework) – это модель описания связанных данных, которая позволяет технологии Семантического веба интерпретировать информацию, представленную в вебе.

Общая идея, на которой основана модель RDF, заключается в следующем: всё, что существует в мире (будь то физический предмет или абстрактное понятие), имеет определенные свойства, а любое свойство имеет конкретные значения. Значит, описать любую сущность можно с помощью элементарных выражений, которые называют эти свойства и их значения.

Основу модели RDF представляет трехчастное утверждение, или триплет, следующего вида:

Субъект – Предикат (или свойство) – Объект (значение свойства).

Триплет можно примерно соотнести с простым предложением вида *Подлежащее – Сказуемое – Дополнение*. Например, утверждение «Книга написана Пушкиным» в RDF-терминологии можно представить следующим образом: субъект – «Книга», предикат – «написана», объект – «Пушкин».

Такое выражение принято представлять в виде графа, в котором субъект и объект – это узлы, а предикат изображается дугой или иной соединительной линией, направленной от субъекта к объекту.



Рис. 1. Триплет RDF

В отношении одного и того же субъекта могут быть составлены и другие выражения, определяющие другие его свойства. Например: утверждение «Книга «Медный всадник», написанная А.С. Пушкиным, издана издательством «Яник» в 2003 г.» можно представить в виде набора триплетов:

Книга – имеет название – «Медный всадник»

Книга – написана автором – А.С.Пушкин

Книга – издана издательством – «Яник»

Книга – издана в году – 2003

Кроме того, объекты (или значения свойств) могут являться субъектами других выражений, образуя значительно более сложные схемы. Множество RDF-утверждений образует ориентированный граф, в котором вершинами являются субъекты и объекты, а рёбра помечены предикатами, например:



Рис. 2. Граф RDF

Выражения в RDF должны быть составлены таким образом, чтобы они могли обрабатываться машинами. Для этого необходимы две вещи:

- система доступных машинной обработке уникальных идентификаторов для обозначения субъекта, предиката и объекта;
- доступный машинной обработке язык для представления выражений и обмена ими между машинами.

Что касается первого, в RDF для идентификации всех ресурсов используются уникальные идентификаторы ресурса – URI (*Uniform Resource Identifier*), а точнее – URIfref (или URIsсылки), т.е. URI вместе с опциональным идентификатором фрагмента в конце. Например, URIfref <http://www.rusmarc.ru/index.html#Section2> состоит из собственно URI и, через символ #, идентификатора фрагмента Section2.

Отметим два момента:

- (1) Термин «ресурс» в RDF используется не в привычном для каталогизаторов значении. Это не объект БЗ, а все, что угодно. Это может быть любая сущность – как информационная (книга, веб-сайт или изображение), так и неинформационная (например, объекты реального мира или некое абстрактное понятие или свойство).
- (2) Для записи URI в RDF принято использовать протокол http, и внешне URI похожи на электронные адреса веб-документов, но в RDF URI используются только для идентификации любого ресурса и не ведут к электронному контенту. URI – лишь уникальное имя, которое идентифицирует ресурс и не несет никакого внутреннего значения.

Субъект и предикат в RDF всегда выражен URIfref; объект может быть выражен либо URIfref, либо постоянной величиной (литералом). Это может быть номер, дата или другая символьная строка. Литералы позволяют включать в RDF текст и используются собственно для описания свойств ресурсов (например, имени автора и дат жизни).

На рис. 3 представлены RDF-утверждения относительно одного и того же ресурса (веб-страницы Национальной службы RUSMARC).

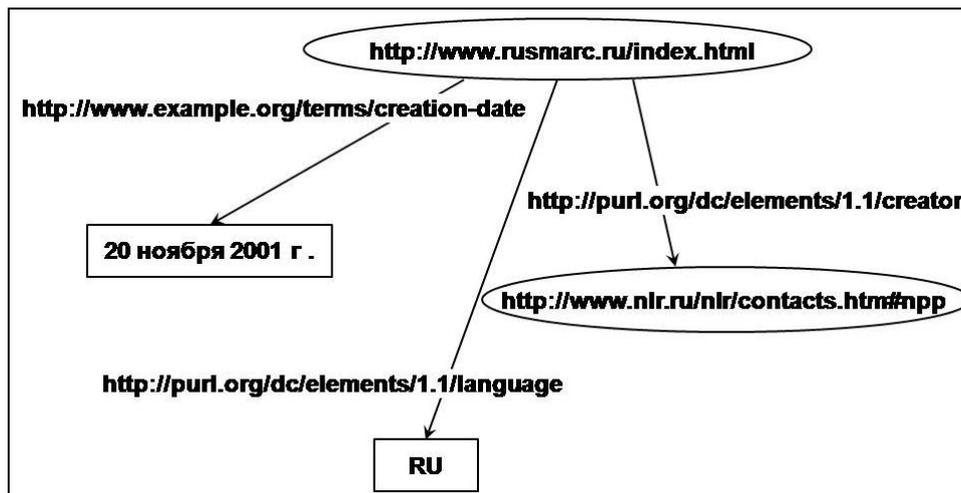


Рис. 3. Граф RDF – описание веб-страницы Национальной службы RUSMARC

Обратим внимание на слайде:

- (1) в качестве универсальных идентификаторов свойств «создатель» и «язык» в этом примере используются теги Dublin Core.
- (2) объекты RDF представлены либо URIref, либо литералами (так, в случае предиката `http://purl.org/dc/elements/1.1/language` объект выражен литералом, который представляет собой международный стандартный двухсимвольный код для обозначения русского языка).

Этот же пример, записанный в виде триплетов, будет выглядеть следующим образом:

```
<http://www.rusmarc.ru/index.html> <http://purl.org/dc/elements/1.1/creator>
    <http://www.nlr.ru/nlr/contacts.htm#npp>
<http://www.rusmarc.ru/index.html> <http://www.example.org/terms/creation-
    date > "20 ноября 2001"
<http://www.rusmarc.ru/index.html> <http://purl.org/dc/elements/1.1/language>
    "RU"
```

Полная нотация RDF выражений иногда может приводить к появлению в записи очень длинных строк. Для удобства записи URI определенным образом сокращаются, используя механизм «пространств имён» XML.

Полные URIref заменяются так называемым *Qualified Name* (или *QName*) – *квалифицированным, или уточненным именем*. *QName* состоит из префикса, определяющего используемое пространство имен, и локальной части имени (локального имени), разделенных двоеточием. Например: если пространству имён Dublin Core `http://purl.org/dc/elements/1.1/` приписан QName-префикс `dc`, тогда URIref `http://purl.org/dc/elements/1.1/creator` может быть записана в виде `dc:creator`.

Применяя QName, получим краткую запись нашего примера:

```
rusmarc:index.html      dc:creator nlr:npp
rusmarc:index.html      exterms:creation-date "20 ноября 2001 г."
rusmarc:index.html      dc:language "RU"
```

RDF – это лишь среда, абстрактная модель, которая определяет способ описания ресурсов с помощью отношений между ними, используя элементарные конструкции *Субъект – Предикат – Объект* и оперируя URI-ссылками вместо слов для обозначения сущностей внутри выражений.

Однако RDF не предоставляет никакого механизма для описания классов сущностей и отношений. Эту задачу выполняют словари – наборы предикатов, которые могут использоваться в том или ином приложении, или онтологии, которые определяет не только сами предикаты, но и типы данных для каждого предиката, а также отношения между предикатами определенной области.

Словарь RDF представляет собой набор URI. Чаще всего для всех терминов словаря выбирается одно пространство имен; как правило, это – пространство имен, которое может контролировать создатель словаря. URIref, которые содержатся в словаре, формируются путем добавления локального имени к URIref словаря. В этом случае словарь представляет собой набор URIref с общим префиксом.

Такой подход RDF использует для определения своего словаря терминов, имеющих специальное значение в RDF. Все URIref в базовом словаре RDF начинаются с <http://www.w3.org/1999/02/22-rdf-syntax-ns#> (Qname-префикс `rdf:`). RDF Schema (RDFS) определяет дополнительный набор терминов, URIref которых начинается с <http://www.w3.org/2000/01/rdf-schema#>, условный префикс – `rdfs:`.

Использование общего префикса – это удобный способ организации словаря для связанных терминов. Однако это лишь соглашение, и ничего более. Модель RDF распознает только полные URIref, она не «заглядывает внутрь» и не анализирует структуру идентификатора. Так, с точки зрения RDF, наличие у группы URIref общего префикса само по себе не означает, что URIref каким-то образом связаны. Более того, нигде не сказано, что URIref с разными префиксами не могут быть частью одного словаря. Создавая словарь для своих целей, вы можете включить в него термины из любого количества существующих словарей. Хотя не стоит забывать, что при этом вы не сможете полностью их контролировать.

В некоторых случаях URI пространства имен словаря используется как URL веб-ресурса, дающего более полную информацию об этом словаре. Так, Qname-префикс `dc:`, ассоциированный с пространством имен <http://purl.org/dc/elements/1.1/>, – это URL словаря Dublin Core, и перейдя по нему, вы получите дополнительную информацию об этом словаре. Однако это тоже лишь соглашение. RDF не предполагает, что URI пространства имен обязательно идентифицирует веб-ресурс.

Для того, чтобы создать RDF-словарь, RDF-документ с описанием терминов необходимо опубликовать в вебе с определенным URI. В дальнейшем такой словарь может использоваться путем ссылок на этот URI в RDF-документе. В каких случаях, для чего и как создавать словари, мы поговорим чуть позже.

Синтаксис RDF

Для того, чтобы опубликовать RDF-графы в вебе, их необходимо представить в последовательной форме, понятной для машин и пригодной для обмена данными (выполнить так называемую сериализацию). Для этого могут использоваться ряд форматов, различающихся конкретным способом записи описания ресурса.

- RDF/XML – выражение графа RDF в виде документа XML;
- RDFa – запись внутри атрибутов произвольного HTML- или XHTML-документа;
- N3 (Notation 3) – краткий способ записи моделей RDF, компактнее и удобнее для чтения, чем XML-запись RDF;
- Turtle – подмножество N3;
- N-Triples – подмножество N3, линейный формат записи

Форматы RDF/XML и RDFa являются стандартом Консорциума W3C.

Пример 1. Описание веб-сайта

Ниже приведен пример описания в RDF/XML веб-страницы Национальной службы RUSMARC, соответствующего RDF-графу на рис. 3.

```
1. <?xml version="1.0"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.   xmlns:exterms="http://www.example.org/terms/"
4.   xmlns:nlr="http://www.nlr.ru/nlr/contacts.htm#"
5.   xmlns:dc="http://purl.org/dc/elements/1.1/">
6.   <rdf:Description rdf:about="http://www.rusmarc.ru/index.html">
7.     <dc:language>RU</dc:language>
8.     <exterms:creation-date>20 ноября 2001 г.</exterms:creation-
9.     date>
10.    <dc:creator
11.      rdf:resource="http://www.nlr.ru/nlr/contacts.htm#npp"/>
12.   </rdf:Description>
13. </rdf:RDF>
```

Строка 1 – XML-декларация, показывающая, что следующий текст есть XML-документ; здесь же указана версия XML.

Строка 2 начинается с элемента `rdf:RDF`. Это показывает, что следующее содержание XML (начинающееся здесь и заканчивающееся конечным тегом в строке 11) предназначено для RDF-представления.

Следующий за `rdf:RDF` элемент в той же строке – декларация пространства имён XML, которая показывает, что все теги этого документа, имеющие префиксы `rdf:`, являются частью пространства имён, идентифицированного указанной URIref – `http://www.w3.org/1999/02/22-rdf-syntax-ns#`, т.е. относятся к базовому RDF-словарю.

Строки 3-5 представляют декларации других пространств имён – в данном случае для префиксов `exterms:`, `nlr:`, `dc:`.

Строки 6-10 – это, собственно говоря, и есть RDF/XML для конкретного выражения, представленного на рис. 3.

Тег `rdf:Description`, открывающий 6-ю строку, показывает начало описания ресурса; далее с помощью атрибута `rdf:about` указывается ресурс, о котором идёт речь (т.е. субъект выражения) – в данном случае веб-сайт `http://www.rusmarc.ru/index.html`.

В строках 7-9 указаны свойства (предикаты) описываемого веб-сайта и их значения.

Строка 7 с тегом `dc:language` описывает свойство (в данном случае – язык) описываемого ресурса. Префикс `dc:` обозначает пространство имен `http://purl.org/dc/elements/1.1/` (см. строку 5), в сочетании с локальным именем `language` мы получаем полную URIref свойства – `http://purl.org/dc/elements/1.1/language`. Содержанием этого элемента свойства является объект выражения – литерал «RU» (код языка). Элемент свойства размещён внутри элемента `rdf:Description`, что показывает, что этот элемент применяется к ресурсу, обозначенному `rdf:about` внутри элемента `rdf:Description`.

Элемент `exterms:creation-date` в строке 8 подобен элементу `dc:language`. Оба элемента представляют величины свойств плоскими литералами и записываются заключением литерала между начальным и конечным тегами XML.

Элемент `dc:creator` в строке 9 представляет собой свойство, величина которого – не литерал, а другой ресурс, идентифицируемый `URIref`. Чтобы показать это различие, элемент `dc:creator` записывается с использованием так называемого тега пустого элемента XML – он не имеет отдельного конечного тега, и величина свойства записывается с использованием атрибута (в данном случае `rdf:resource`) внутри этого пустого элемента. Поскольку `URIref` используется как величина атрибута, RDF/XML требует, чтобы она была записана полностью без использования `QName`, в отличие от того, как это делается для обозначения наименований элементов и свойств.

Строка 10 отмечает конец элемента `rdf:Description`.

Наконец, строка 11 отмечает конец элемента `rdf:RDF`, открытого в строке 2.

Пример 2. Описание электронного журнала

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/">
  <rdf:Description rdf:about=
    "http://www.dlib.org/dlib/may98/05contents.html">
    dc:title>DLIB Magazine - The Magazine for Digital Library
      Research - May 1998</dc:title>
    <dc:description>D-LIB magazine is a monthly compilation of
      contributed stories, commentary, and
      briefings.</dc:description>
    <dc:contributor>Amy Friedlander</dc:contributor>
    <dc:publisher>Corporation for National Research
      Initiatives</dc:publisher>
    <dc:date>1998-01-05</dc:date>
    <dc:type>electronic journal</dc:type>
    <dc:format>text/html</dc:format>
    <dc:identifier rdf:resource="urn:issn:1082-9873"/>
    <dc:subject>
      <rdf:Bag>
        <rdf:li>library use studies</rdf:li>
        <rdf:li>magazines and newspapers</rdf:li>
      </rdf:Bag>
    </dc:subject>
    <dcterms:isPartOf rdf:resource="http://www.dlib.org"/>
  </rdf:Description>
</rdf:RDF>
```

Первые строки представляют декларации пространств имён.

Тег `rdf:Description` показывает начало описания ресурса; сам ресурс указывается с помощью атрибута `rdf:about` – в данном случае описывается журнал.

Далее указаны свойства описываемого журнала и их значения:

- название (`dc:title`)
- описание (аннотация) (`dc:description`)
- соисполнитель (`dc:contributor`)
- издатель (`dc:publisher`)
- дата (`dc:date`)
- тип ресурса (`dc:type`)

- формат (dc:format)
- идентификатор (dc:identifier)
- предмет и ключевые слова (dc:subject)¹

Кроме того, в примере используется квалификатор DublinCore isPartOf (из отдельного словаря dcterms:.) для указания того, что описываемый журнал есть часть веб-ресурса <http://www.dlib.org>.

Пример 3. Описание статьи в электронном журнале

В следующем примере описывается отдельная статья в журнале.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/">
  <rdf:Description
    rdf:about="http://www.dlib.org/dlib/may98/miller/
    05miller.html">
    <dc:title>An Introduction to the Resource Description
      Framework</dc:title>
    <dc:creator>Eric J. Miller</dc:creator>
    <dc:description>The Resource Description Framework (RDF) is
      an infrastructure that enables the encoding, exchange
      and reuse of structured metadata. rdf is an
      application of xml that imposes needed structural
      constraints to provide unambiguous methods of
      expressing semantics. rdf additionally provides a
      means for publishing both human-readable and machine-
      processable vocabularies designed to encourage the
      reuse and extension of metadata semantics among
      disparate information communities. the structural
      constraints rdf imposes to support the consistent
      encoding and exchange of standardized metadata
      provides for the interchangeability of separate
      packages of metadata defined by different resource
      description communities. </dc:description>
    <dc:publisher>Corporation for National Research
      Initiatives</dc:publisher>
    <dc:subject>
      <rdf:Bag>
        <rdf:li>machine-readable catalog record
          formats</rdf:li>
        <rdf:li>applications of computer file organization
          and access methods</rdf:li>
      </rdf:Bag>
    </dc:subject>
    <dc:rights>Copyright © 1998 Eric Miller</dc:rights>
    <dc:type>Electronic Document</dc:type>
    <dc:format>text/html</dc:format>
    <dc:language>en</dc:language>
```

¹ Для представления предметных рубрик здесь используется RDF-контейнер Bag (т.е. ресурс, имеющий тип *rdf:Bag*), представляющий группу ресурсов или литералов, для которой не имеет значения их порядок. Имя элемента *rdf:li* здесь служит мнемоникой для "list item" из HTML.

```
<dcterms:isPartOf
  rdf:resource="http://www.dlib.org/dlib/may98/
  05contents.html"/>
</rdf:Description>
</rdf:RDF>
```

Первые строки представляют декларации пространств имён.

Описываемый ресурс указан с помощью атрибута `rdf:about` элемента `rdf:Description` – в данном случае статья в журнале.

Далее указаны свойства статьи и их значения:

- название (`dc:title`)
- автор (`dc:creator`)
- описание (аннотация) (`dc:description`)
- информацию об авторском праве (`dc:rights`)
- издатель (`dc:publisher`)
- предмет и ключевые слова (`dc:subject`)
- тип ресурса (`dc:type`)
- формат (`dc:format`)
- язык (`dc:language`)
- квалификатор `isPartOf` из словаря `dcterms:`, который в этом случае показывает, что данная статья есть часть журнала.

Словари RDF

Во всех приведённых примерах для описания журнала и статьи в дополнение к базовым словарям RDF использован словарь Dublin Core. Возможность использования нескольких различных словарей в документе RDF – очень полезное свойство. Но что делать, если существующие словари не позволяют адекватно описать нужный нам набор данных? В этом случае RDF предусматривает возможность создания и собственного словаря средствами *RDFS*, *OWL* (Web Ontology Language) и *SKOS* (Simple Knowledge Organization System).

1. RDFS

RDFS (RDF Schema) – язык для описания словарей, или упрощённых онтологий, в RDF, предоставляющий механизмы для описания групп взаимосвязанных ресурсов и отношений между ресурсами.

Примитивы языка RDFS определяются в двух пространствах имен:

- <http://www.w3.org/1999/02/22-rdf-syntax-ns#> (префикс `rdf:`)
- <http://www.w3.org/2000/01/rdf-schema#> (принят префикс `rdfs:`)

Основными структурными элементами RDFS являются классы и свойства, построенные на ограниченном словаре RDF.

Классы

- `rdfs:Resource` – включает все ресурсы – всё, что может быть описано в RDF.
- `rdfs:Class` – декларирует, что ресурс является классом для других ресурсов.
- `rdf:Property` – класс всех свойств RDF.

Свойства

Свойства в RDF также являются ресурсами, входят в класс `rdf:Property` и описывают отношения между ресурсом-субъектом и ресурсом-объектом.

Обратим внимание, что RDF-словарь определяет не классы с точки зрения, какие свойства они могут иметь, а наоборот – основными элементами словаря являются именно свойства, и они определяются с точки зрения классов ресурсов, к которым могут применяться эти свойства. Сравните, например, модель FRBR, в которой определены прежде всего основные объекты, а затем уже атрибуты, которые могут относиться к этим объектам.

Некоторые свойства в RDFS:

- `rdfs:domain` – домен свойства субъекта, используется для указания класса, который может иметь данное свойство;
- `rdfs:range` – используется для указания, какие значение может иметь то или иное свойство; например, мы можем определить, что свойство `eg:author` имеет домен `eg:Document` и диапазон `eg:Person`.
- `rdf:type` – свойство, используемое для указания, что ресурс является экземпляром класса;
- `rdfs:subClassOf` – показывает, что все частности одного класса являются частностями другого класса (например, класс *собаки* – подкласс класса *млекопитающие*); таким образом можно декларировать иерархии классов;
- `rdfs:subPropertyOf` – показывает, что все ресурсы, связанные одним свойством, также связаны и другим свойством. Например, свойство *длина* – есть подсвойство свойства *размер*.

Классы и свойства сами по себе являются ресурсами, которые идентифицируются URI и публикуются в вебе, как и другие ресурсы, в виде RDF-документов.

На рис. 4 приведен фрагмент опубликованного словаря RDFS. Для примера рассмотрим описание свойства `rdfs:Subclass`.

```
- <rdf:RDF>
  <owl:Ontology rdf:about="http://www.w3.org/2000/01/rdf-schema#" dc:title="The RDF Schema vocabulary (RDFS)"/>
  - <rdfs:Class rdf:about="http://www.w3.org/2000/01/rdf-schema#Resource">
    <rdfs:isDefinedBy rdf:resource="http://www.w3.org/2000/01/rdf-schema#" />
    <rdfs:label>Resource</rdfs:label>
    <rdfs:comment>The class resource, everything.</rdfs:comment>
  </rdfs:Class>
  - <rdfs:Class rdf:about="http://www.w3.org/2000/01/rdf-schema#Class">
    <rdfs:isDefinedBy rdf:resource="http://www.w3.org/2000/01/rdf-schema#" />
    <rdfs:label>Class</rdfs:label>
    <rdfs:comment>The class of classes.</rdfs:comment>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource" />
  </rdfs:Class>
  - <rdf:Property rdf:about="http://www.w3.org/2000/01/rdf-schema#subClassOf">
    <rdfs:isDefinedBy rdf:resource="http://www.w3.org/2000/01/rdf-schema#" />
    <rdfs:label>subClassOf</rdfs:label>
    <rdfs:comment>The subject is a subclass of a class.</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class" />
    <rdfs:domain rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class" />
  </rdf:Property>
  - <rdf:Property rdf:about="http://www.w3.org/2000/01/rdf-schema#subPropertyOf">
    <rdfs:isDefinedBy rdf:resource="http://www.w3.org/2000/01/rdf-schema#" />
    <rdfs:label>subPropertyOf</rdfs:label>
    <rdfs:comment>The subject is a subproperty of a property.</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
    <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
  </rdf:Property>
```

Рис. 4. Фрагмент словаря RDFS

В элементе `<rdf:Description>` указывается описываемый ресурс – в данном случае свойство `subclassOf`.

Элемент `<Rdfs:comment>` – это текстовое описание ресурса (Субъект является подклассом класса).

Элемент `<Rdfs:range>` – диапазон свойства, `<Rdfs:domain>` – домен свойства (в данном случае и субъект, и объект предиката `subclassOf` должны быть классами, т.е. `subclassOf` – это отношение, которое может возникать между двумя классами).

Решая задачу представления данных в RDF, мы должны прежде всего не изобретать новые словари, а использовать термины из уже существующих, распространенных словарей. В таком случае приложения, работающие со стандартными словарями, смогут принять наши данные без дополнительной настройки.

В сообществе Семантического веба уже разработаны и широко используются многочисленные словари, назовем лишь некоторые из них, наиболее актуальные для наших задач и наиболее распространенные:

- **RDF** – базовый словарь RDF.
- **RDFS** – RDF-схема.
- **OWL (Ontology Web Language)** – Язык сетевых (или веб-) онтологий; основан на стандартных словарях RDF и RDFS, но определяет больше классов, что позволяет придавать больше смысла RDF-предикатам.

Словари RDFS и OWL используются в случаях, когда необходимо отразить отнесение терминов к определенному классу (например, что все кошки являются млекопитающими).

- **SKOS (Simple Knowledge Organization System)** – словарь для представления таксономий и слабо структурированных знаний. Если RDFS и OWL предоставляют словари для описания концептуальных моделей в терминах классов и их свойств, SKOS – это словарь для выражения иерархий понятий. SKOS широко используется для представления тезаурусов и систем ПР.
- **FOAF (Friend-of-a-Friend)** – словарь для описания людей, их деятельность и отношения с другими людьми и объектами.
- **Dublin Core** – определяет общие атрибуты метаданных.
- **BIBO (Bibliographic Ontology)**
- **SIOC (Semantically-Interlinked Online Communities)**
- **(DOAP) Description of a Project**
- **Music Ontology**

В случае если существующие словари не позволяют адекватно описать конкретный набор данных, необходимо разработать новые термины в специализированном словаре с применением средств RDFS и OWL.

При этом следует учитывать следующие правила:

1. Дополнять уже существующие словари новыми терминами, а не изобретать заново то, что уже изобрели до вас.
2. Определять новые термины только в том пространстве имен, которое вы контролируете. RDF не запрещает использовать и другие пространства имен, однако в случае если вы создадите URI в пространстве имён, которое вы не контролируете, вы не сможете гарантировать, что больше никто не будет использовать тот же URI для обозначения чего-то другого.

3. Устанавливать связи между новыми терминами и терминами существующих словарей, используя средства RDFS и OWL. Само понятие связанных данных основано на связях, очень важно при определении новых терминов привязать их к терминам стандартных словарей. В противном случае стандартные браузеры просто не смогут интерпретировать ваши данные. Стандартные средства для установления такого соответствия – свойства `rdfs:subClassOf` и `rdfs:subPropertyOf`.
4. Снабжать каждый новый термин текстовым названием и комментариями (используя свойства `rdfs:label` и `rdfs:comment`, специально для этого и предназначенные)
5. Определять только то, что действительно необходимо, не перегружая словарь излишними деталями. Например, указание доменов и диапазонов свойств может быть полезным для определения, как следует использовать то или иное свойство, но в некоторых случаях может привести к неправильным логическим выводам.

В заключение хотелось бы вернуться к тому, с чего мы начинали – говоря о представлении библиографических данных в RDF, мы должны четко понимать, какую цель мы преследуем и что хотим получить в итоге. Нам нужно не просто разобрать наши данные на элементарные тройки, построить RDF-графы, перевести их в RDF/XML или другой формат сериализации и гордо заявить, что мы в семантическом вебе. Самое главное – сделать это так, чтобы наши данные были не просто доступны, но и понятны остальному вебу. Для этого, прежде чем решать техническую задачу, необходимо решить задачу интеллектуальную – подготовить модели данных и словари, соблюдая правила, о которых мы говорили выше. Эти модели должны быть, с одной стороны, стандартными и понятными для стандартных поисковых машин, а с другой – в нужной степени специфичными, чтобы отразить специфику библиографических данных.