

## ОСНОВЫ RDF/XML

---

По материалам



<http://www.w3schools.com/rdf/>



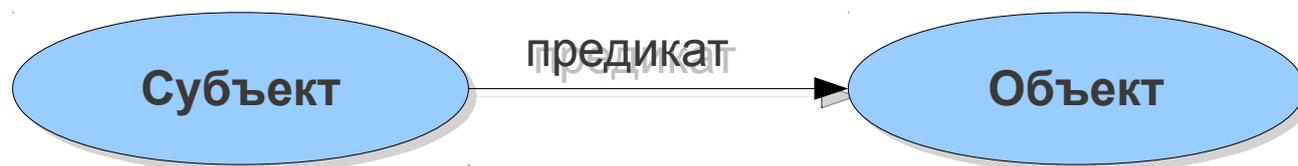
<http://www.w3.org/rdf/>

[ Тузовский А.Ф., Чириков С.В., Ямпольский В.З.  
Системы управления знаниями (методы и  
технологии). - Томск: Изд-во НТЛ, 2005 ]

RDF (Resource Description Framework) — абстрактная модель

<http://www.w3.org/RDF/>

Согласно этой модели знания представляются тройкой в виде отношения:



Таким образом, в модели RDF узлами графа являются **субъекты** и **объекты**.

**Ребро** имеет направление всегда **на объект**.

**Предикат** часто также называется (выполняет роль) **свойством**.

Каждый **объект** и **субъект** могут задаваться как URI, так и символьно, а также для некоторых задач могут быть пустыми

**предикат** должен иметь уникальный **идентификатор URI** (URL)

Предикат имеет свое описание, идентифицируемое URI, с целью однозначного «понимания» (например, системами семантического анализа).

## Основы XML:

XML — Extensible *Markup Language* — расширяемый язык разметки.

<http://www.w3.org/TR/REC-xml/>

Предназначен для хранения структурированной информации для обмена данными между программами, а также для организации разметки документов.

Визуально структура XML-файла представляется в виде дерева элементов, записанных строго определенным образом, некоторой предметной области.

Элементы в XML описываются тегам:

- Парные: `<открывающий_тег>Содержимое элемента</закрывающий_тег>`
- Одиночные (пустые элементы): `<одиночный_тег />`

Каждый тег может иметь свойства, описание которых имеет следующий формат:

- `<тег свойство1="значение" свойство2="значение">Содержимое элемента</тег>`
- `<тег свойство1="значение" свойство2="значение" />`

Начало документа начинается с объявления (декларации) XML, содержащего номер версии, кодировки, а также различных зависимостей, например:

```
<?xml version="1.0" encoding="UTF-8"?>
```

В XML допускается только один корневой элемент.

## Пример файла XML:

```
<?xml version="1.0" encoding="utf-8"?>
<keyboard vendor="Genius">
  <led>
    <led1 title="Num Lock" color="green" />
    <led2 title="Caps Lock" color="green" />
  </led>
  <functional>
    <fkeys>
      <f1 title="F1" fn="sleep" />
      <f2 title="F2" fn="wifi" />
      <f3 title="F3" fn="outscreen" />
      <f4 title="F4" fn="bright_dec" />
      <f5 title="F5" fn="bright_inc" />
    </fkeys>
    <tab title="Tab" />
    <caps title="Caps Lock" />
    <shiftkeys>
      <lshift title="Shift" />
      <rshift title="Shift" />
    </shiftkeys>
    <ctrlkeys>
      <lctrl title="Ctrl" />
      <rctrl title="Ctrl" />
    </ctrlkeys>
  </functional>
</keyboard>
```

Одни и те же понятия на языке XML можно описать различными способами:

...

```
<сотрудник>  
  <фιο>Иванов Петр Семенович</фιο>  
</сотрудник>
```

...

```
<сотрудник>  
  <фιο>  
    <фамилия>Иванов</фамилия>  
    <имя>Петр</имя>  
    <отчество>Семенович</отчество>  
  </фιο>  
</сотрудник>
```

...

```
<сотрудник фιο="Иванов Петр Семенович" />
```

В XML представлении информации может возникнуть неоднозначность (человеческий фактор)

Язык XML часто используется как формат обмена данными. В этом случае все стороны, принимающие участие в обмене должны согласиться на общую структуру (DTD или XML Schema)

XML используется в качестве языка сериализации других языков (Semantic Web).

При разработке XML-приложений могут быть использованы совпадающие имена для атрибутов и элементов — словари. Часто одни приложения используют имена из других приложений. Для различения имен различных приложений была разработана спецификация пространств имен для XML — XML Namespace Specification

`<dc:creator>`

dc — префикс пространства имен Dublin Core, creator — имя элемента .

Указание пространств(а) имен в XML:

`<организация xmlns = "kms.cctpu.edu.ru/ontology" xmlns:dc = "http://purl.org/dc/elements/1.1/" />`

XML не может выразить смысл тегов. Программы, работающие с документами XML требуют определенные наборы тегов, согласующиеся с определенными стандартами.

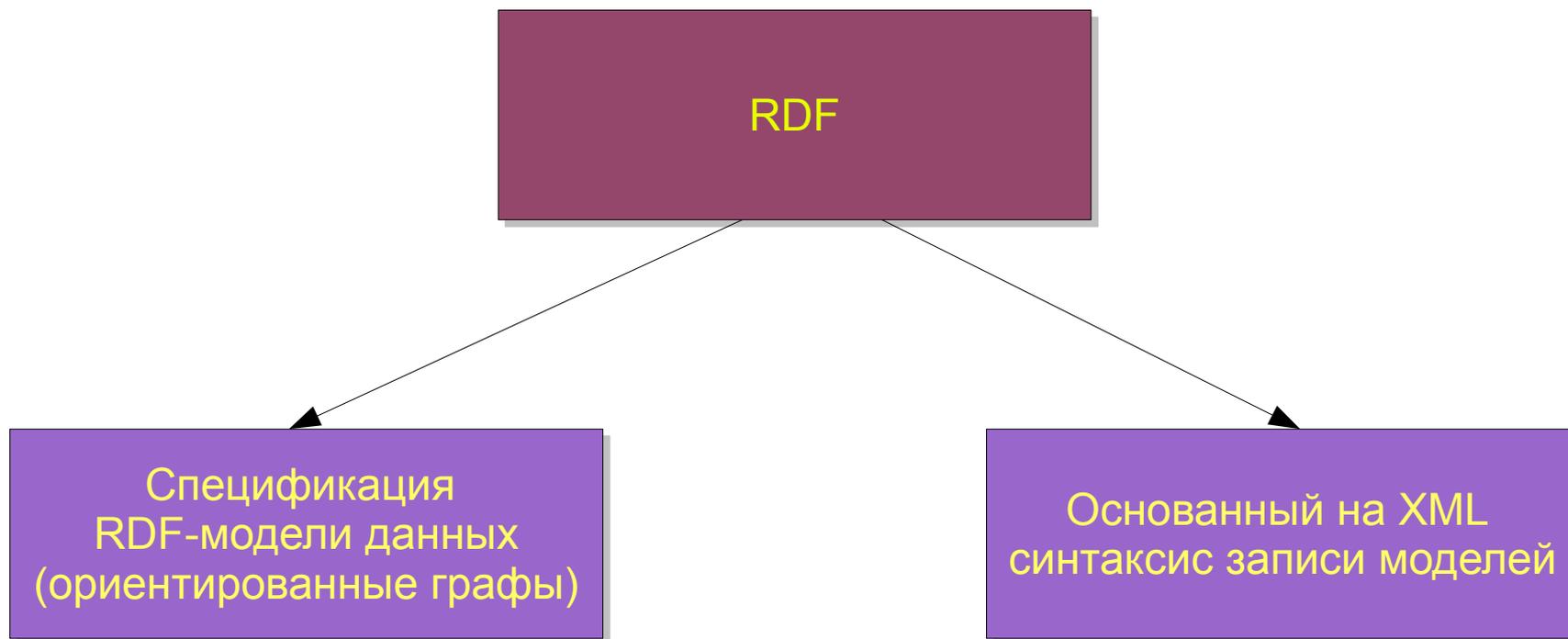
Для облегчения программной обработки данных XML позволяет определить грамматику тегов. Эта информация содержится в определении типа документа — Document Type Definition (DTD).

Конкретное описание на языке DTD определяет ограничения на структуру XML-документа.

```
<!ELEMENT сотрудник (имя, должность, датарождения)>  
<!ELEMENT фио (#PCDATA)>  
<!ELEMENT должность (#PCDATA)>  
<!ELEMENT датарождения (#PCDATA)>
```

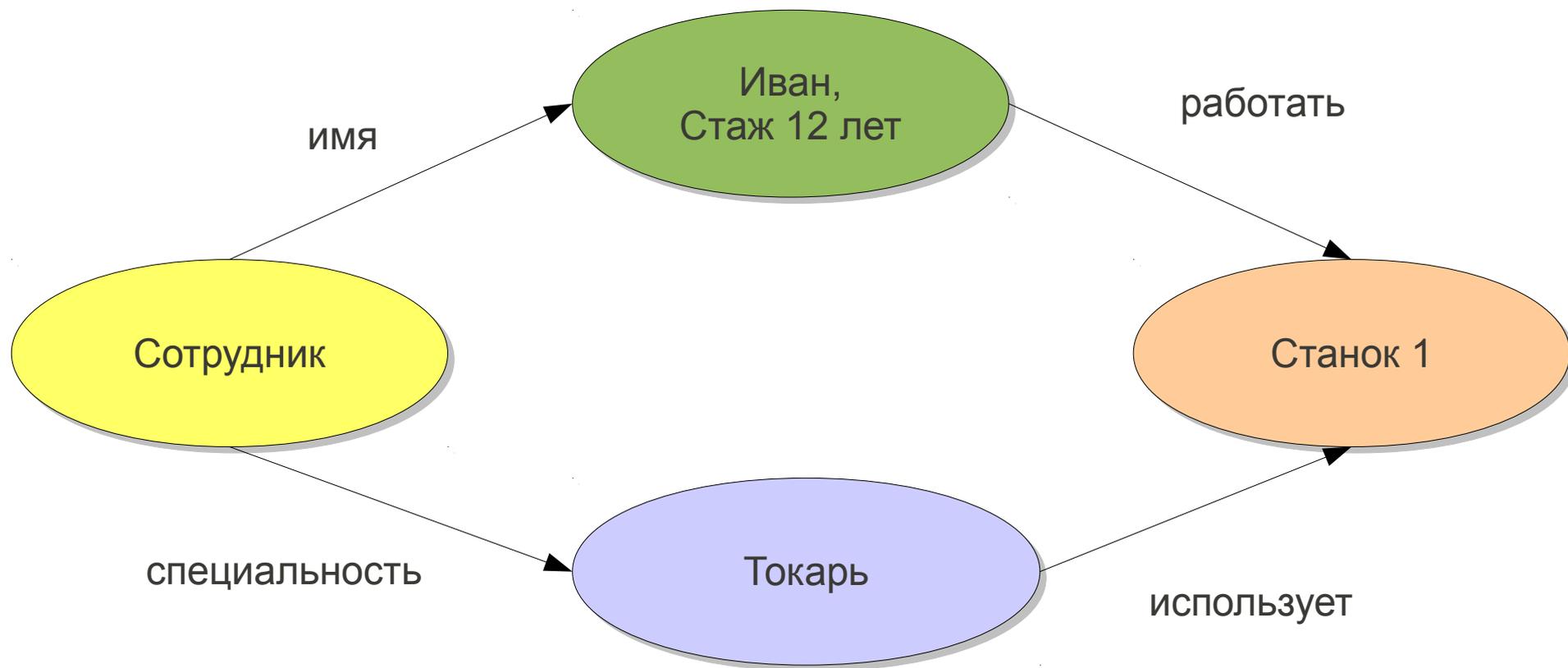
## XML Schema

```
...  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:element name="сотрудник">  
    <xs:complexType>  
      <xs:sequence>  
        <xs:element name="имя" re="xs:string"/>  
        <xs:element name="должность" re="xs:string"/>  
        <xs:element name="датаРождения" re="xs:date"/>  
      </xs:sequence>  
    </xs:complexType>  
  </xs:element>  
</xs:schema>  
...
```



Основными строительными элементами являются:

- Ресурсы (URI)
- Литералы



Данный граф может быть представлен в виде перечня ребер, каждое из которых записывается в виде тройки. Это модель данных языка RDF

(сотрудник, имя, Иван)  
(сотрудник, специальность, токарь)  
(Иван, работать, станок1)  
(токарь, использует, станок1)

На языке XML этот граф может быть записан в виде

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:kms="http://kms.cctpu.edu.ru/ontology#">
```

```
  <rdf:Description about="kms:сотрудник">  
    <kms:имя>kms:Иван</kms:имя>  
  </rdf:Description>
```

```
  <rdf:Description about="kms:сотрудник">  
    <kms:специальность rdf:resource = "kms:токарь" />  
  </rdf:Description>
```

```
  <rdf:Description about="kms:Иван">  
    <kms:работать rdf:resource = "http://kms.cctpu.edu.ru/ontology#Станок1" />  
    <kms:стаж rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">12</kms:стаж>  
  </rdf:Description>
```

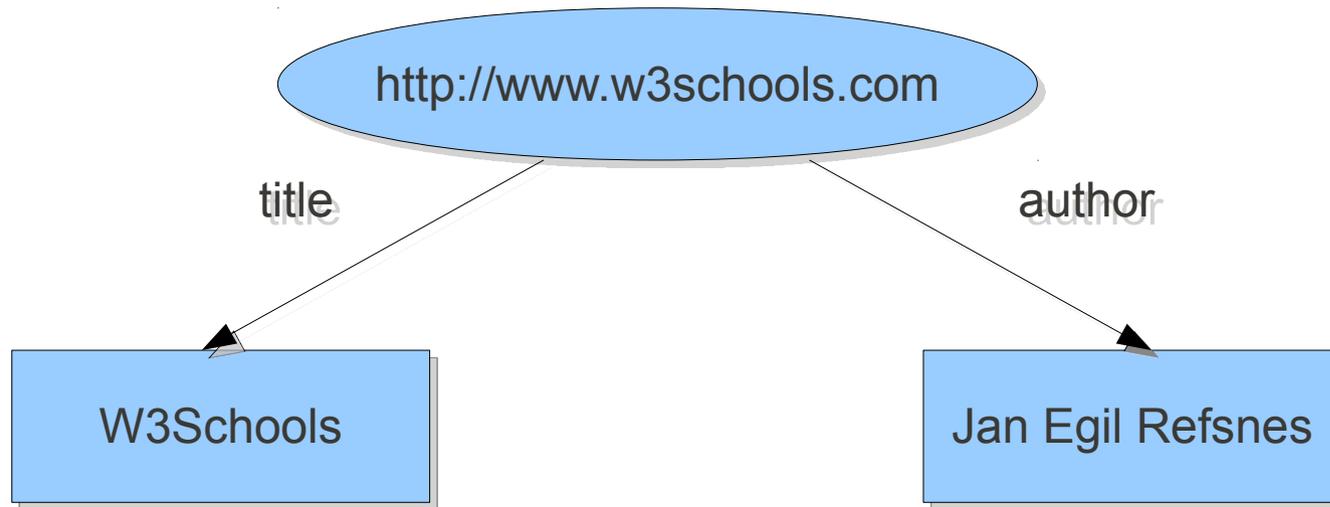
```
  <rdf:Description about="kms:Иван">  
    <kms:работать rdf:resource = "kms:Станок1">  
  </rdf:Description>
```

```
</rdf:RDF>
```

Основными элементами пространства имен rdf являются:

- `rdf:RDF` – элемент, определяющий начало rdf-описания;
- `rdf:Description` – элемент, с которого начинается описание каждого ребра графа;
- `rdf:about` – атрибут (для `rdf:Description`), указывает на ресурс, который будет описываться в элементе `rdf:Description`;
- `rdf:ID` – атрибут (для `rdf:Description`), указывает на ID ресурса, который будет описываться в элементе `rdf:Description`;
- `rdf:datatype` – атрибут в элементе свойства, который указывает, что значение свойства имеет тип (обычно XML Schema: `rdf:datatype="xsd:decimal"`, где `xsd` – префикс пространства имен XML Schema);
- `rdf:type` – предопределенное свойство, указывающее, что ресурс относится к определенному классу, заданному в онтологии.
- `rdf:Resource` – атрибут, который показывает, что значением свойства является другой ресурс, обозначаемый URI;
- `rdf:value` – задает основное значение свойства, если значением свойства является структура;
- `rdf:parseType` – атрибут свойства, который указывает, что содержание элемента должно интерпретироваться специальным способом

Пример описания знаний о web-странице в виде некоторых ее свойств:



В формате XML:

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:si="http://www.w3schools.com/rdf/">
```

```
  <rdf:Description rdf:about="http://www.w3schools.com">
    <si:title>W3Schools</si:title>
    <si:author>Jan Egil Refsnes</si:author>
```

```
  </rdf:Description>
```

```
</rdf:RDF>
```

Пояснения:

Заголовок XML-файла:

```
<?xml version="1.0"?>
```

Признак описания модели RDF:

```
<rdf:RDF
```

«Подключение» пространств имен через определение префиксов (пользователь):

Определяем префикс «rdf» - основной синтаксис RDF/XML:

«Description», «about», «resource», etc.

```
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

Определяем префикс «si» - некоторое определенное пространство имен, где описаны свойства «title», «author»

```
xmlns:si="http://www.w3schools.com/rdf/">
```

Описание субъекта (Description) как ресурса в сети («about»)

```
<rdf:Description rdf:about="http://www.w3schools.com">
```

Описание предикатов (свойств) с помощью пространств имен и объектов

Предикат «title», описанный в пространстве имен, обозначенном как «si»

```
<si:title>W3Schools</si:title>
```

Предикат «author», из пространства имен, обозначенном как «si»

```
<si:author>Jan Egil Refsnes</si:author>
```

```
</rdf:Description>
```

```
</rdf:RDF>
```

Объект предметной области, имеющий URI с описанием, часто задается пустым элементом XML, т. е. «непарным» тегом:

```
<dc:author rdf:resource="http://www.books.fake/authors/id0012/" />
```

Где свойство «rdf:resource» тега описано в пространстве имен синтаксиса RDF (<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)

Как было сказано, субъект и объект могут быть представлены без URI (символьно).

В предыдущем примере так были описаны объекты «автор» и «заголовок», относящиеся к субъекту «веб-страница».

Символьное определение может иметь и субъект, например:

```
<rdf:Description ex:fullname="Dave Beckett">
  <ex:homepage rdf:resource="http://purl.org/net/dajobe/" />
</rdf:Description>
```

где, предикаты «fullname» и «homepage» определены в пространстве имен, определенном префиксом «ex», обозначающим ссылку на ресурс, где они описаны.

## Пространства имен (описание предикатов)

Как было показано, подключаются в корневом элементе документа `<rdf:RDF>`:  
`<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">`

Фрагмент файла `http://www.w3.org/1999/02/22-rdf-syntax-ns#` :

```
<rdf:Property rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">
  <rdfs:isDefinedBy rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
  <rdfs:label>type</rdfs:label>
  <rdfs:comment>The subject is an instance 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#Resource" />
</rdf:Property>
```

```
<rdfs:Class rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property">
  <rdfs:isDefinedBy rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
  <rdfs:label>Property</rdfs:label>
  <rdfs:comment>The class of RDF properties.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource" />
</rdfs:Class>
```

```
<rdfs:Class rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement">
  <rdfs:isDefinedBy rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
  <rdfs:label>Statement</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource" />
  <rdfs:comment>The class of RDF statements.</rdfs:comment>
</rdfs:Class>
```

**Пространство имен DC (Dublin Core)** — «Дублинское ядро» - формат метаданных для описания широкого круга ресурсов (как электронных, так и физических)  
<http://dublincore.org/documents/dcq-rdf-xml/>

Пример подключения: `<rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/">`

dc:creator

dc:contributor

dc:publisher

dc:subject

dc:description

```
<rdf:Property rdf:about="http://purl.org/dc/elements/1.1/creator">
```

```
<rdfs:label xml:lang="en-US">Creator</rdfs:label>
```

dc:identifier

```
<rdfs:comment xml:lang="en-US">An entity primarily responsible for making the resource.</rdfs:comment>
```

dc:relation

```
<dcterms:description xml:lang="en-US">Examples of a Creator include a person, an organization, or a service. Typically, the name of a Creator should be used to indicate the entity.</dcterms:description>
```

dc:source

```
<rdfs:isDefinedBy rdf:resource="http://purl.org/dc/elements/1.1/">
```

dc:rights

```
<dcterms:issued>1999-07-02</dcterms:issued>
```

dc:format

```
<dcterms:modified>2008-01-14</dcterms:modified>
```

dc:type

```
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
```

dc:title

```
<dcterms:hasVersion rdf:resource="http://dublincore.org/usage/terms/history/#creator-006"/>  
<skos:note xml:lang="en-US">A second property with the same name as this property has been declared in the dcterms: namespace (http://purl.org/dc/terms/). See the Introduction to the document "DCMI Metadata Terms" (http://dublincore.org/documents/dcmi-terms/) for an explanation.</skos:note>
```

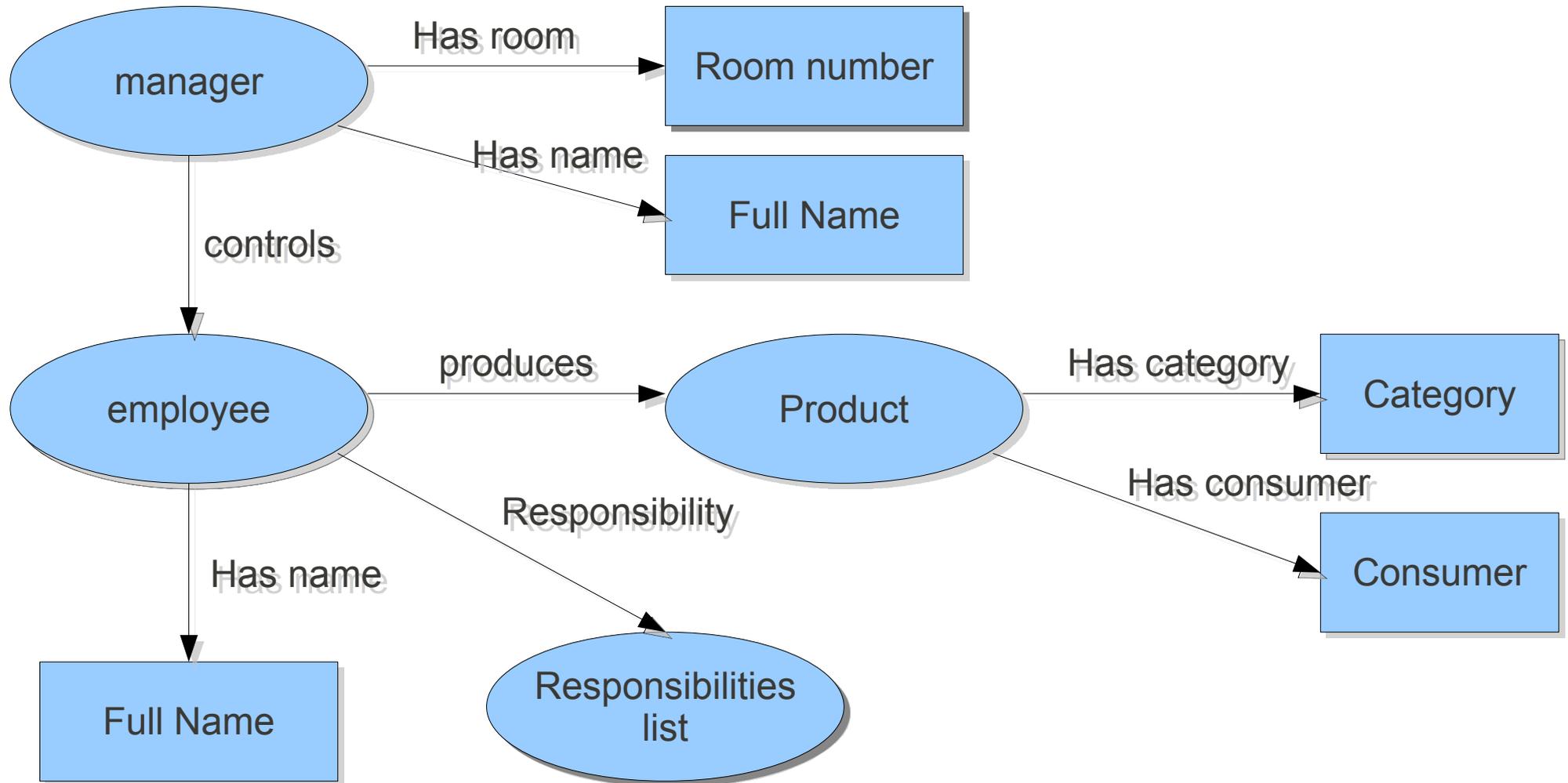
dc:date

```
</rdf:Property>
```

dc:coverage

dc:language

# Пример описания абстрактной модели



Один из способов запись модели в формате XML:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:work="http://www.work.org/company/rdf/">
```

```
  <rdf:Description about="http://www.work.org/company/personal/0006/">
    <work:position>Начальник</work:position>
    <work:room>7</work:room>
    <work:fullname>Полиграф Полиграфович</work:fullname>
    <work:controls rdf:resource="http://www.work.org/company/personal/0018/" />
  </rdf:Description>

  <rdf:Description about="http://www.work.org/company/personal/0018/">
    <work:position>Наемный рабочий</work:position>
    <work:fullname>Петр Иванович</work:fullname>
    <work:responsibility rdf:resource="http://www.work.org/company/resp/002/" />
    <work:produces rdf:resource="http://www.work.org/company/products/0739/" />
  </rdf:Description>

  <rdf:Description about="http://www.work.org/company/products/0739/">
    <work:category>Слесарный инструмент</work:category>
    </work:consumer>Слесари</work:consumer>
  </rdf:Description>
```

```
</rdf:RDF>
```