Arbeitsgruppe
CSW
Corporate Semantic Web

Freie Universität Berlin

Web-based Information Systems

Semantic Web I

Semantic Web Stack, RDF, SPARQL

Prof. Dr. Adrian Paschke

Arbeitsgruppe Corporate Semantic Web (AG-CSW)
Institut für Informatik, Freie Universität Berlin
paschke@inf.fu-berlin
http://www.inf.fu-berlin/groups/ag-csw/

Introduction

Example - Literature Search in the HCLS Publication Database PubMed.org

- **>20.000.000 literature abstracts**
  - 2000-5000 new publications per day / yearly about 500.000
- Excellent source if you know what you are looking for

CSW
Corporate Semantic Web

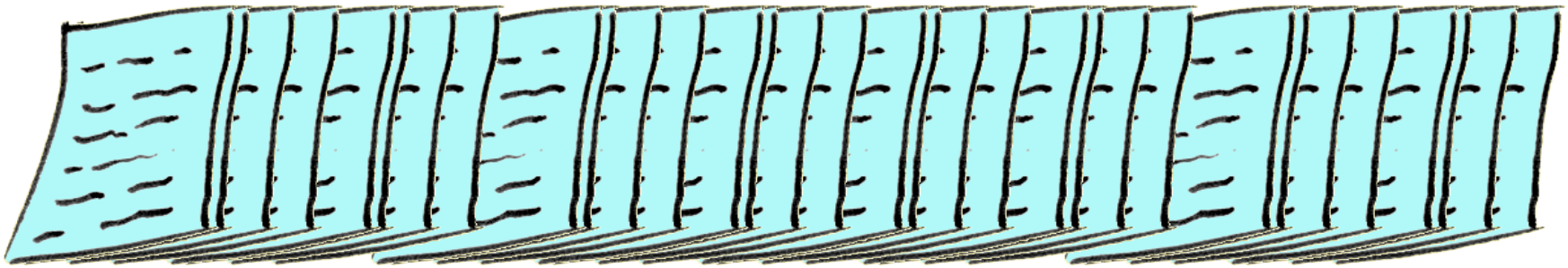
Example Queries

- *What are the leading authors and institutions in liver transplantation?*
- *Which diseases are related to the HIV virus?*
- *Which anatomical structure is affected by the „helicobacter pylori“ bacterium?*

Problem: Keyword-based Literature Search (pure syntactic search)

Your PubMed search: „Levamisole inhibitor"

long result list in PubMed



Results from PubMed Publication Database

Title

Author

Journal

Year

- Lorenz P, Transcriptional repression ediated by the KRAB domain of the human C2H2 zinc finger protein Kox1/ZNF10 does not require histone deacetylation. Biol Chem. 2001 Apr;382(4):637-44.
- Fre ericks WJ. An engineered PAX3-KRAB t nal represso the malignant phenotype of alveolar rhabdomyosarcoma cells harboring the endogenous PAX3-FKHR Mol Cell Biol. 2000 Jul;20(14):5019-31.

However, for a machine things look different!

Corporate Semantic Web

Results from PubMed Publication Database

- ☹□❒♏■⌘ ⌦✍ ❄❒♋■⬩♍❒♓□♦♓□■♋● ❒♏□❒♏⬩⬩♓□■
  ❍♏♎♓♋♦♏♎ ♌♲ ♦♒♏ 😐☼✌👌 ♎□❍♋♓■ □➚ ♦♒♏ ♒◆❍♋■ 👍🗎🖗🗎
  ⌘♓■♍ ➚♓■♑♏❒ □❒□♦♏♓■ 😐□☒🗁✍☪☠☞🗁🗀 ♎□♏⬩ ■□♦
  ❒♏❑◆♓❒♏ ≈
  👌♓□● 👍♒♏❍ **Solution: Tags (XML)**
- ☞❒♏♎♏❒♓♍&
  ♦❒♋■⬩♍❒♓□♦ ♋●♓♑■♋■♦
  □♒♏■□♦⌧□♏ ❍♋ ♍♏●●⬩
  ♒♋❒♌□❒♓■♑ ♑□♑♏■♏✍
  💣□● 👍♏●● 👌♓□●✍ 🗎🗀🗀🗀 ☺◆●🖴🗎🗀☎🗁🗐◑💻🖫🗀🗁↜✍🗏🗁.

  …

Results from PubMed Publication Database

- <author>☹□❒♏■⌘ ➱</author><title>❄❒♋■⬧♍❒♓□◆♓□■♋● ❒♏□❒♏⬧⬧♓□■ ❍♏♎♓♋♦♏♎ ♌🖂 ♦♒♏ 😐☼✌👌 ♎□❍♋♓■ □➚ ♦♒♏ ♒◆❍♋■ 👍🗎🖗🗎 ⌘♓■♍ ➚♓■♑♏❒ □❒□♦♏♓■ 😐□⊠🗁📣☪☠☞🗁🗀 ♎□♏⬧ ■□♦ ❒♏❑◆♓❒♏ ♒♓⬧♦□■♏ ♎♏♋♍♏♦🖂●♋♦♓□■. </title> <journal>👌♓□● 👍♒♏❍ </journal><year>🗎🗀🗀🗁<year>
- <author>☹□❒♏■⌘ ➱</author><title>❄❒♋■⬧♍❒♓□◆♓□■♋● ❒♏□❒♏⬧⬧♓□■ ❍♏♎♓♋♦♏♎ ♌🖂 ♦♒♏ 😐☼✌👌 ♎□❍♋♓■ □➚ ♦♒♏ ♒◆❍♋■ 👍🗎🖗🗎 ⌘♓■♍ ➚♓■♑♏❒ □❒□♦♏♓■ 😐□⊠🗁📣☪☠☞🗁🗀 ♎□♏⬧ ■□♦ ❒♏❑◆♓❒♏ ♒♓⬧♦□■♏ ♎♏♋♍♏♦🖂●♋♦♓□■. </title> <journal>👌♓□● 👍♒♏❍ </journal><year>🗎🗀🗀🗁<year>
- ...

However, for a machine things look different!

CSW
Corporate Semantic Web

Results from PubMed Publication Database

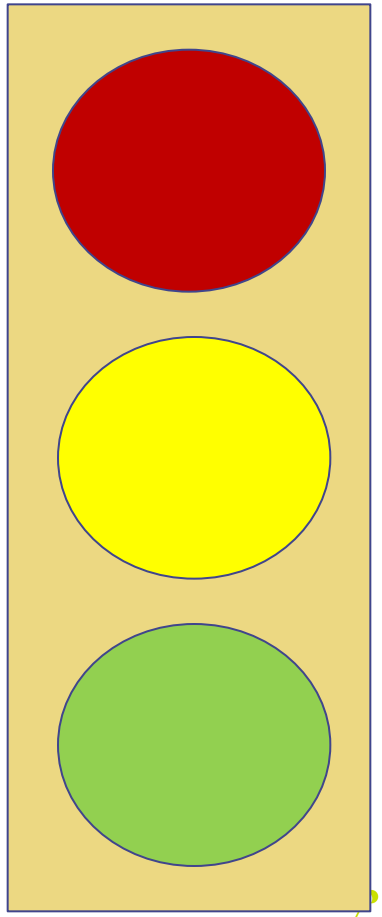
- [illegible]
- [illegible]
- …

Solution:
Use Ontologies
(Semantics)

CSW
Corporate Semantic Web

Example: Traffic Light
Syntax – Semantics - Pragmatics

- **Syntax**
  - *green (bottom); yellow; red*
- **Semantics**
  - *green = go; …; red = stop*
- **Pragmatics**
  - If ***red*** and ***no traffic***
    then ***allowed to go***



Example - XML Syntax vs. Semantics

Adrian Paschke is a lecturer of Logic Programming

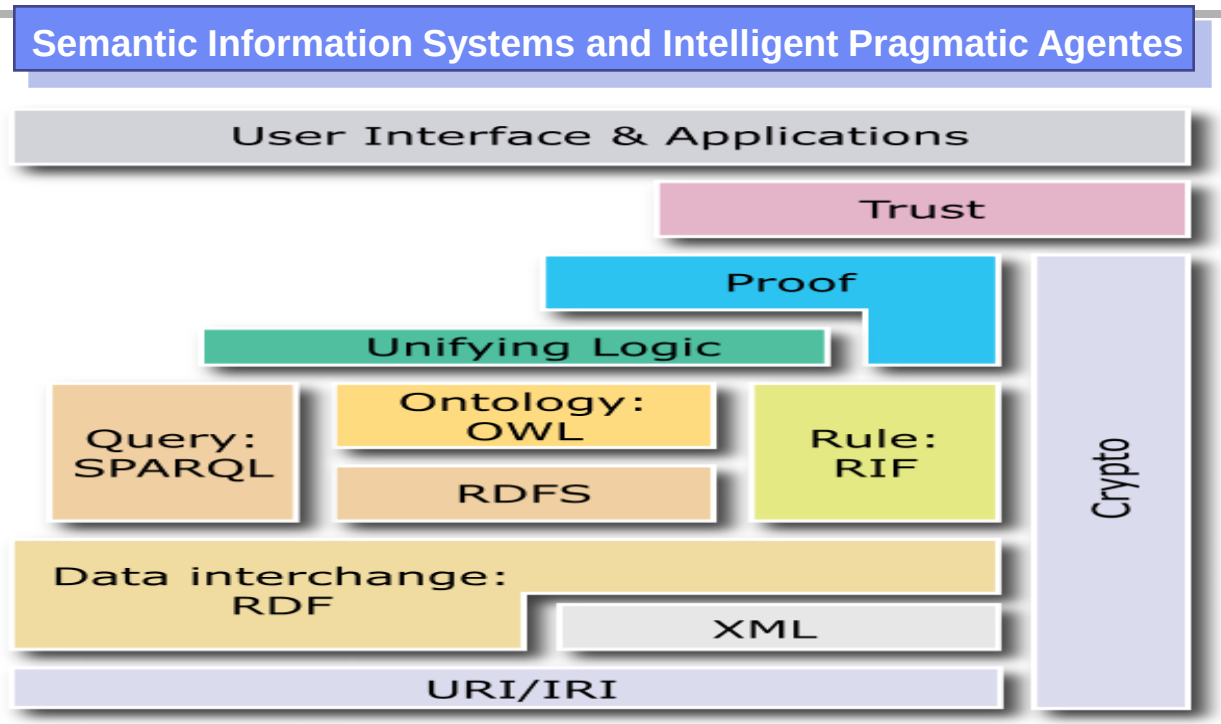
```
<course name=“Logic Programming">
        <lecturer>Adrian Paschke</lecturer>
</course>

<lecturer name=“Adrian Paschke">
        <teaches>Logic Programming</teaches>
</lecturer>
```

Opposite nesting (syntax), same meaning (semantics)!

Syntax – Semantics - Pragmatics

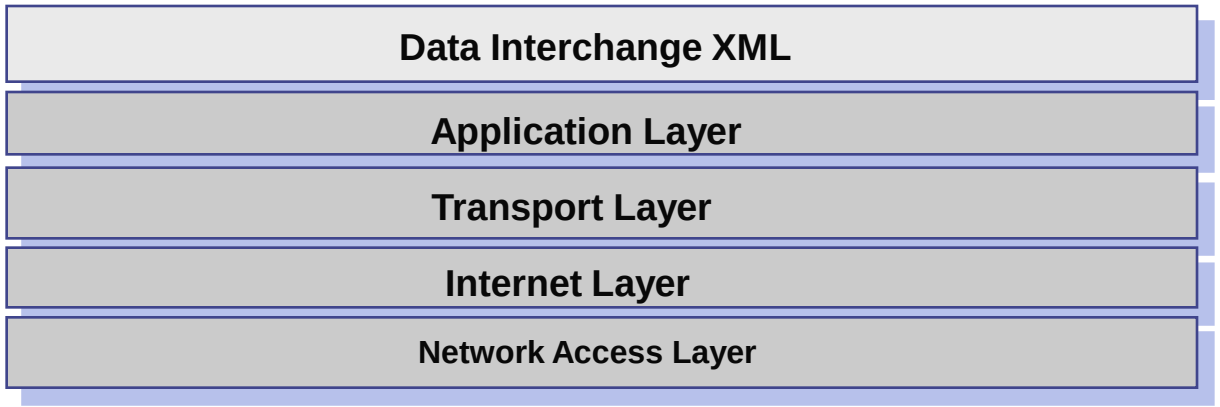
- **Syntax**
  - about **form**
- **Semantics**
  - about **meaning**
- **Pragmatics**
  - about **use**.

Overview – Semantic Web



Corporate Semantic Web / Pragmatic Web

Semantic Web Technologien



Internet Technologien:

XML, DTD, XSD, DOM, SAX, …

FTP, SMTP, HTTP, TELNET, NEWS

TCP, UDP

IP (mit ICMP, ARP)

SLIP, PPP,
IEEE 802.3, 802.11 …

CSW
Corporate Semantic Web

Semantic Web I

- Semantic Web – An Introduction (Part 1)
  - Semantic Web Vision
  - RDF
  - RDF Query Languages / SPARQL

Lecture is partially based on Grigoris Antoniou, Frank van Harmelen: Semantic Web Primer

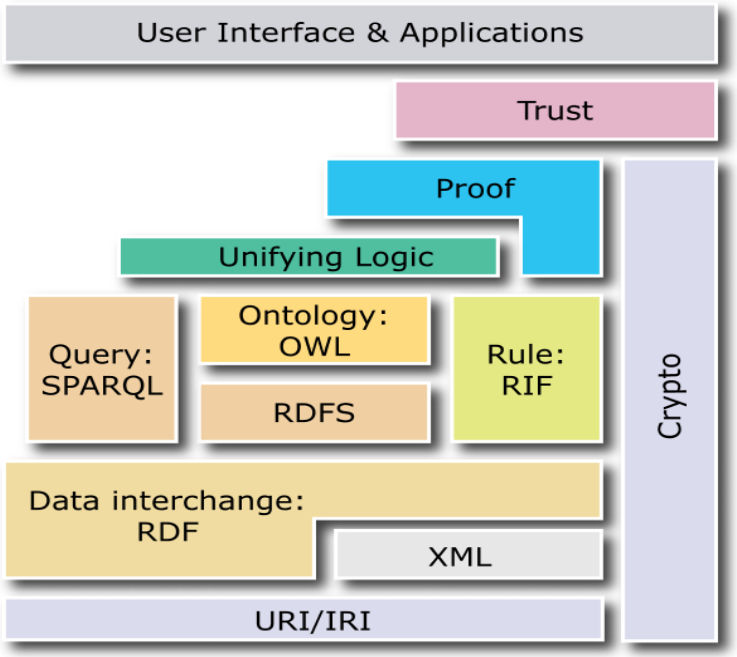
Corporate Semantic Web

The Semantic Web Vision

Semantic Web – An Introduction

- "The Semantic Web is an **extension of the current web** in which information is given **well-defined meaning**, better enabling **computers and people to work in cooperation**."
  - *Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web*

- „Make the Web understandable for machines“



W3C Stack 2007

Corporate Semantic Web

The Current Web

- Web contents are typically designed for human use
  - Automatically generated web contents, e.g. from databases, are presented without any semantics and without information about the original structure.
- Missing support for automated processing via tools
  - e.g. often only keyword-based search engines (e.g. sorting based on Google page rank)

Example: Problems with Keyword-based Search

- Many results (high recall), but low accuracy (low precision)
- or few or no results (low recall)
- Results are single website (URIs)
- **Results need to be interpreted by humans**
- Search results are **not directly usable by automated software tools**
  - Semantic information about the meaning of web contents are missing

Drawbacks of XML

- XML is a universal meta language for defining markup
- It provides a uniform framework for interchange of data and metadata between applications
- However, XML does not provide any means of talking about the semantics (meaning) of data
- E.g., there is no intended meaning associated with the nesting of tags
  - It is up to each application to interpret the nesting.

Nesting of Tags in XML

Adrian Paschke is a lecturer of Web Based Information Systems

<course name="Web Based Information Systems">
<lecturer>Adrian Paschke</lecturer>
</course>

<lecturer name="Adrian Paschke">
<teaches>Web Based Information Systems</teaches>
</lecturer>

Opposite nesting, same information!

The Semantic Web Approach

- Representation of web contents in a machine-readable format
  - Annotation with **metadata** and **ontologies**
- Usage of intelligent inference techniques (l***ogic and inference***) in order to process web contents automatically and derive new knowledge from existing one
- Automated tools, e.g. ***rule-based Expert Systems, Web Services, Software Agents***

➔ The Semantic Web is an extension of the existing WWW

CSW
Corporate Semantic Web

Building Blocks of the Semantic Web (and beyond)

1. Explicit Metadata on the WWW
2. Ontologies
3. Logic and Inference
4. Software Agents and Semantic Web Services

1. Explicit Metadata on the Web

- Metadata are data about data
- Metadata on the Web:
  - *Machine processable information about information on the Web*
  - examples e.g.: PICS, Dublin Core, RDF, IEEE LOM (Learning Objects Metadata), FOAF, …
- Problem domains:
  - Syntax:
    - Which representation and interchange format for metadata?
  - Semantics:
    - Which metadata are allowed for resources (metadata vocabulary, schema)
  - Association problem:
    - How to connect metadata with resources (who defines the metadata, are metadata separated from the content, etc.)

2. Ontologies

- “An **ontology** is an explicit specification of a conceptualization “ T. Gruber
- Ontologies described the common knowledge of a domain (semantics):
  - → Semantics interoperability between (connected) vocabularies
- Typical components:
  1. Classes (concepts) of the domain
  2. Properties (roles) of the classes
  3. Constraints
  4. Individuals (instances) of classes

3. Logic and Inference

- Logic is a discipline concerned with the principles of inference and reasoning
- Formal languages for the representation of knowledge with clear semantics
  - Declarative knowledge representation:

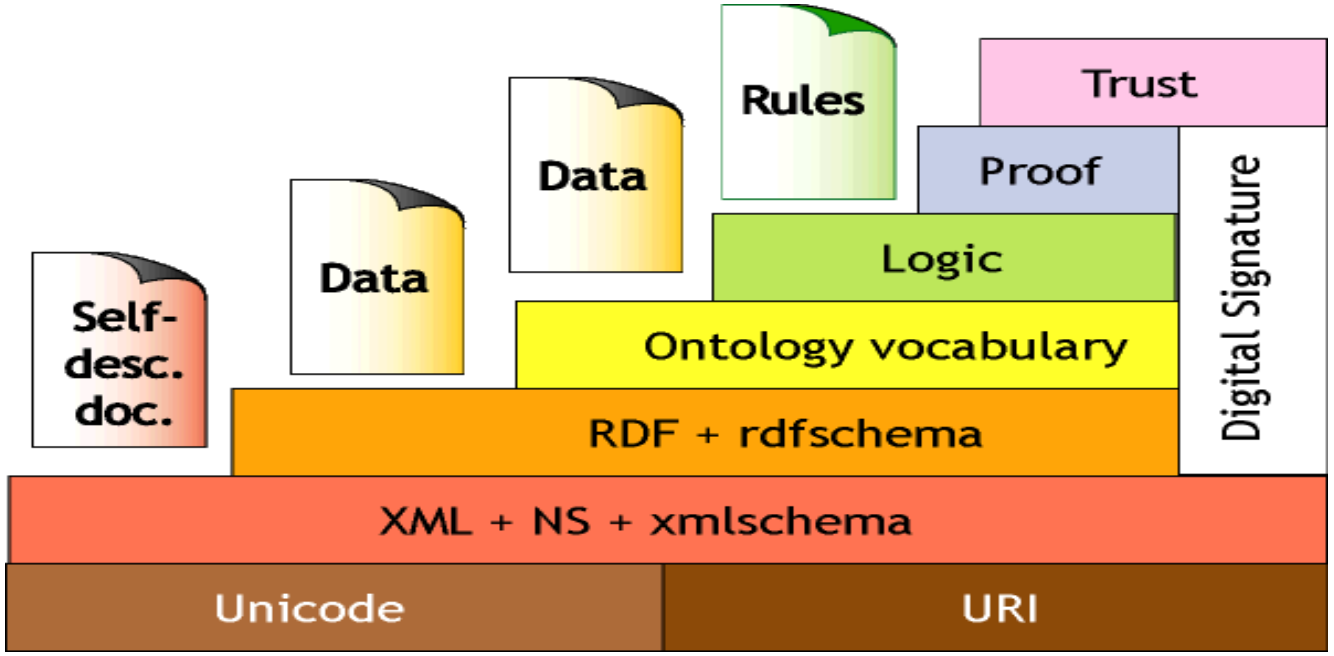
    express what is valid, the responsibility to interpret this and to decide on how to do it is delegated to an interpreter / reasoner
- Automated reasoner, e.g., a rule engine, can derive conclusions from given knowledge (inference)

4. Software Agents and Web Services

- **Intelligent Software Agents** act autonomously and pro-active
  - They have an internal knowledge base with decision/reaction logic (e.g. rule-based expert systems)
  - Examples: Personal agents (e.g. Rule Responder), search robots
- **Web Service**
  - In general: any IT service provided on the Web
  - “A '***Web service'*** (also **Web Service**) is defined by the W3C as "a software system designed to support interoperable Machine to Machine interaction over a network." Web services are frequently just Web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services.” (Wikipedia)
- => no clear separation between web agents and web services (in the broad sense)
  - but level of self-autonomous decisions is higher in web agents

CSW
Corporate Semantic Web

Layered Architecture of the Semantic Web



W3C Stack 2003

- Principles (Original Semantic Web Stack as of 2003)
  - Development in layers – each layer depends on the other
  - Downwards compatible
  - Up-wards: partial understanding
  - **But:** New stack proposals exists

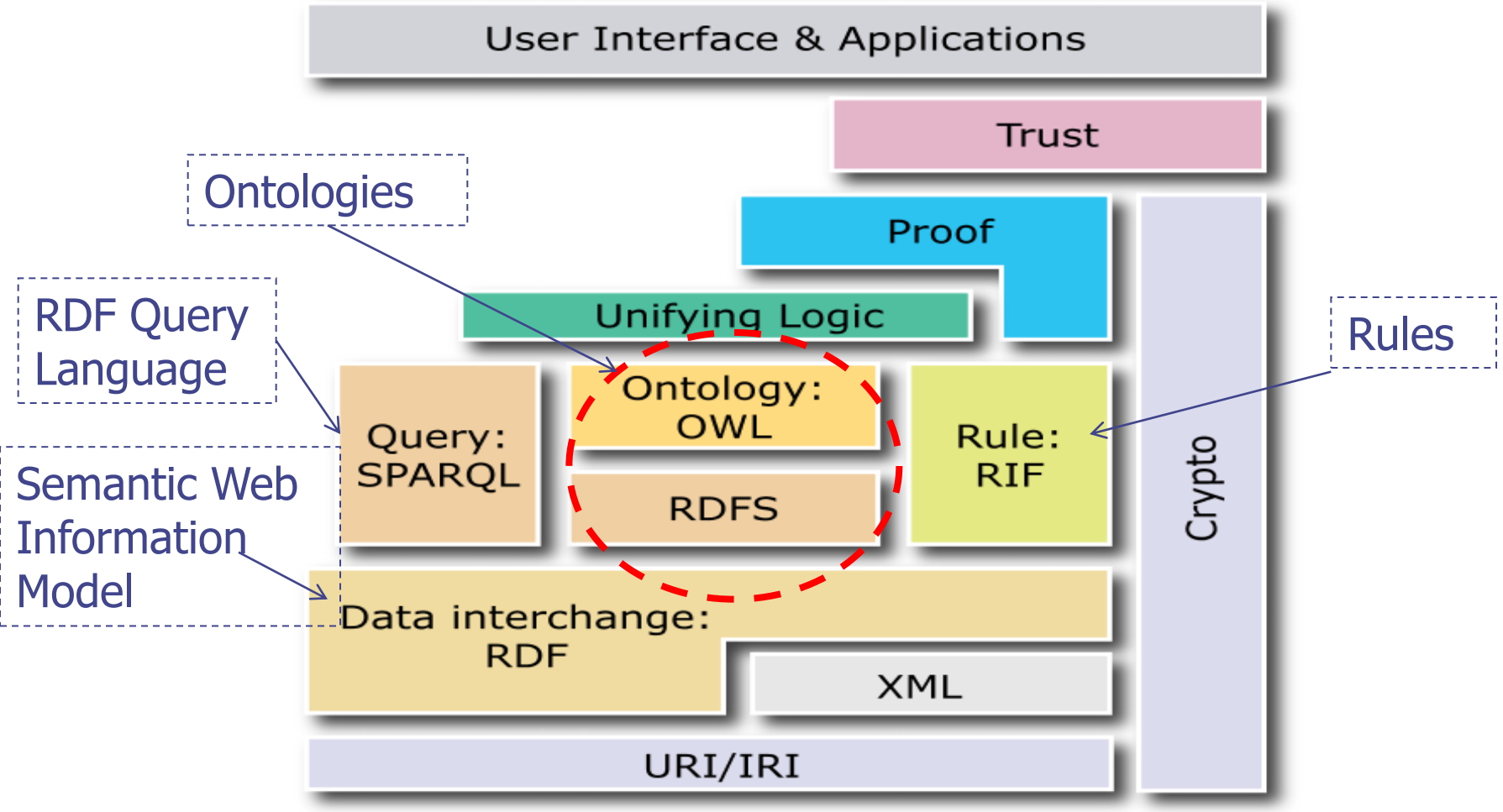
Overview on the Semantic Web Layers (1)

- XML Layer
  - Syntactic basis
- RDF Layer
  - RDF as data model for facts and metadata
  - RDF schema (RDFS) as simple ontology language (mainly taxonomies)
- Ontology Layer
  - Expressive ontology languages
  - Actual standard: Web Ontology Language (OWL)

Overview on the Semantic Web Layers (2)

- Logic Layer
  - Extension of the ontology languages, e.g. with rules
- Proof Layer
  - Generation of proofs-, interchange of proofs, validation
- Trust Layer
  - Digital signatures
  - recommendations, ratings

The (current) W3C Semantic Web Architecture



W3C Semantic Web Stack since 2007

RDF

Resource Description Framework

RDF

- WWW consists of networked resources
- Resource Description Framework (RDF) – a standard for the **description of resources**
- **Metadata** and other structures can be represented in RDF
  - e.g Dublin Core, FOAF, iCal RDF
- RDF can be represented in XML as **RDF/XML**
  - but in general, is independent from XML
- RDF can be embedded in HTML as **RDFa**
- RDF (triples) can be stored in **Triplestores** (RDF databases)

RDF History

- Start:
  - Extension of the PICS system (Platform for Internet Content Selection)
- 22.2. 1999: W3C Recommendation „RDF Model & Syntax Specification“
- 27.3.2000: W3C Recommendation „RDF Schema Specification 1.0“
- 25.9.2001: W3C Working Draft „RDF Model Theory“
- Today: e.g. proposal for extensions with negation

RDF Data Model

- RDF data consists of triples:
  - *Nodes* (**resources** – arbitrary URI)
  - Attributes (**properties**)
  - Values
  - Basic building block: **resource-attribute-value** triple = RDF statement
- Values again can be nodes
- The RDF information maps to a directed and labeled graph
- **RDF/XML** can be used for interchange of RDF models
  - XML namespaces can avoid naming conflicts

Basic Ideas of RDF (2)

- The fundamental concepts of RDF are:
  - **resources**
  - **properties**
  - **statements**

Resources

- We can think of a resource as an object, a “thing” we want to talk about
  - E.g. authors, books, publishers, places, people, hotels
- Every resource has a **URI**, a Universal Resource Identifier (or IRI)
- A URI can be (see lecture on WWW)
  - a URL (Web address) or
  - some other kind of unique identifier

Properties

- **Properties** are a special kind of resources
- They describe relations between resources
  - E.g. “written by”, “age”, “title”, etc.
- Properties are also identified by URIs
- Advantages of using URIs:
  - A global, worldwide, unique naming scheme
  - Reduces the homonym problem of distributed data representation

Statements

- Statements assert the properties of resources
- A statement is an resource-attribute-value triple
  - It consists of a resource, a property, and a value
- Values can be **resources** or **literals**
  - Literals are atomic values (strings, integer…)

Three Views of a Statement

- A **triple**
- A **piece of a graph**
- A **piece of XML code**

Thus an RDF document can be viewed as:

- A set of triples
- A graph (semantic net)
- An XML document

Statements as Triples

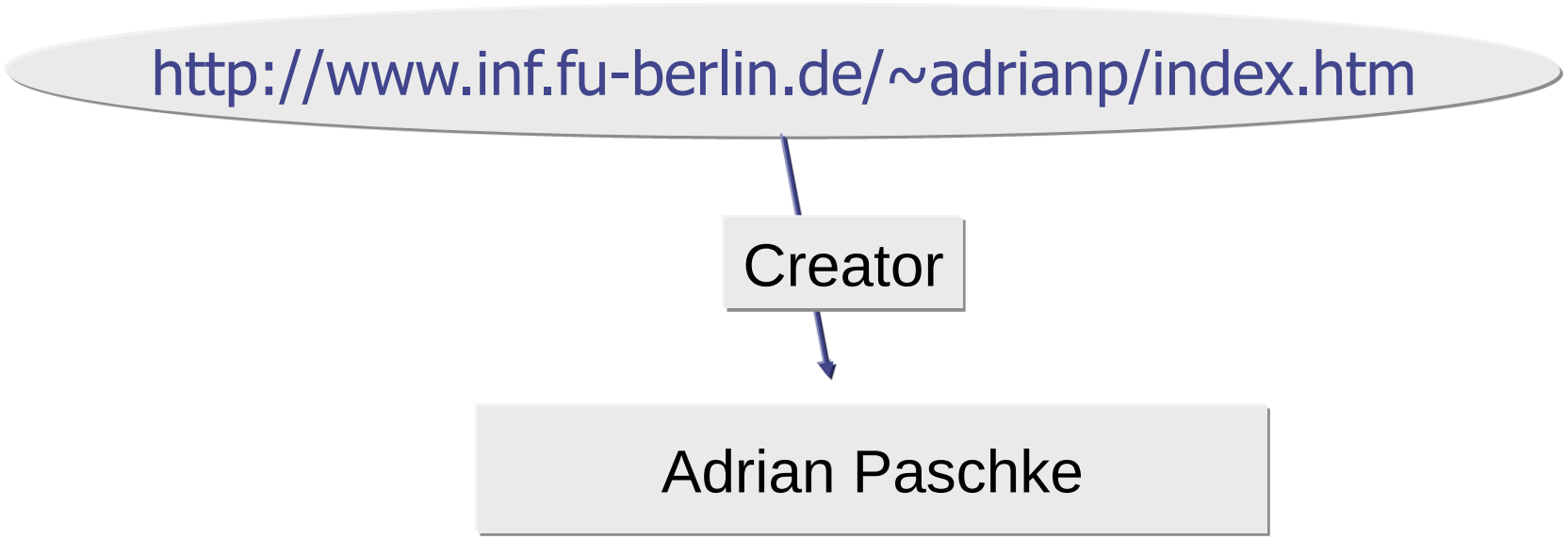
(“Adrian Paschke”,

http:// www.inf.fu-berlin.de /site-owner,

http://www.inf.fu-berlin.de/adrianp/**)**

- The triple *(x,P,y)* can be considered as a logical formula *P(x,y)*
  - Binary predicate P relates object x to object y
  - RDF offers only binary predicates (properties)

First RDF diagram



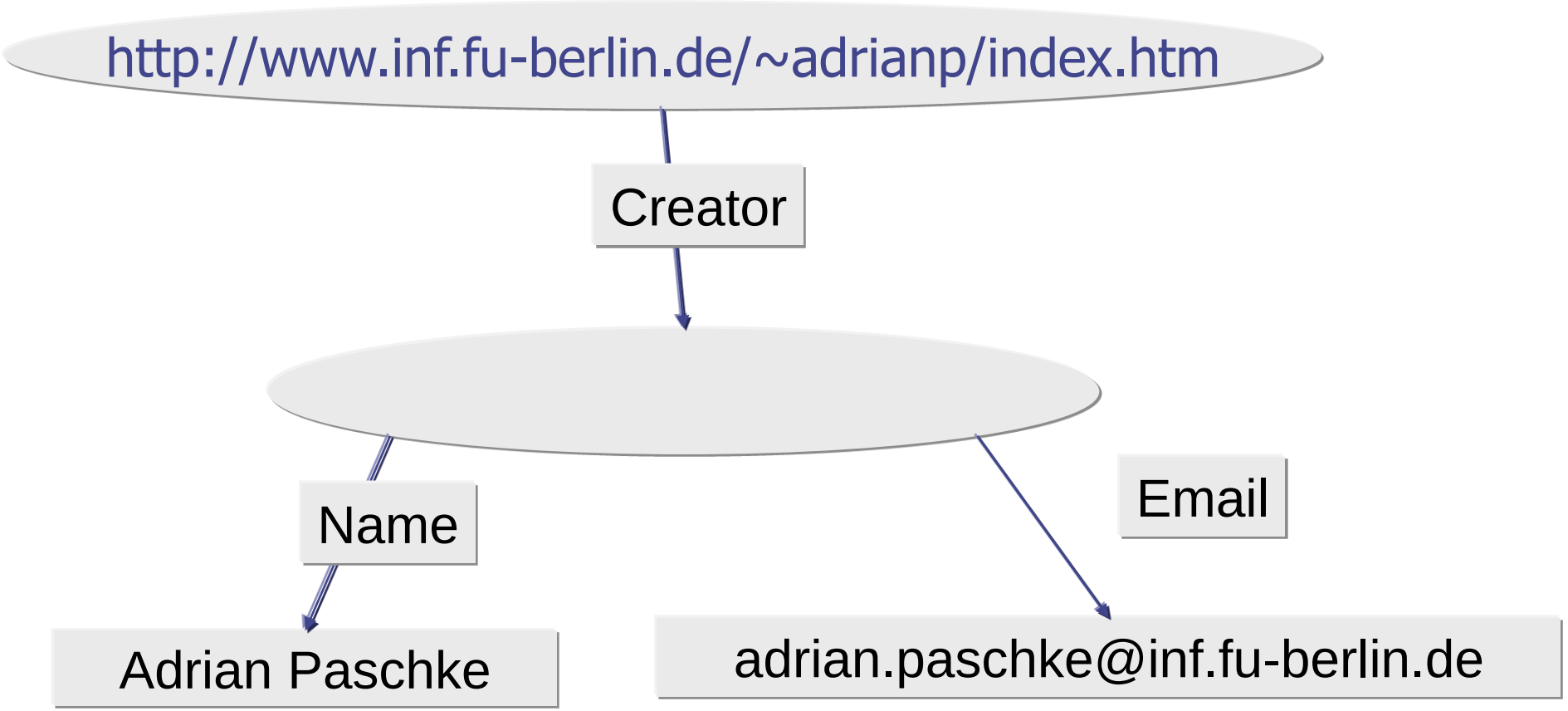
Subject (= Ressource): http://www.inf.fu-berlin.de/~adrianp/index.htm
Predicate (= Property Attribute): Creator
Object (= Value): Adrian Paschke

Read: <Ressource> has <Property> <Value>

RDF/XML Version

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:c="http://description.org/schema/">

  <rdf:Description about="http://www.inf.fu-berlin.de/~adrianp/index.htm">
    <c:Creator>Adrian Paschke</c:Creator>
  </rdf:Description>

</rdf:RDF>
```

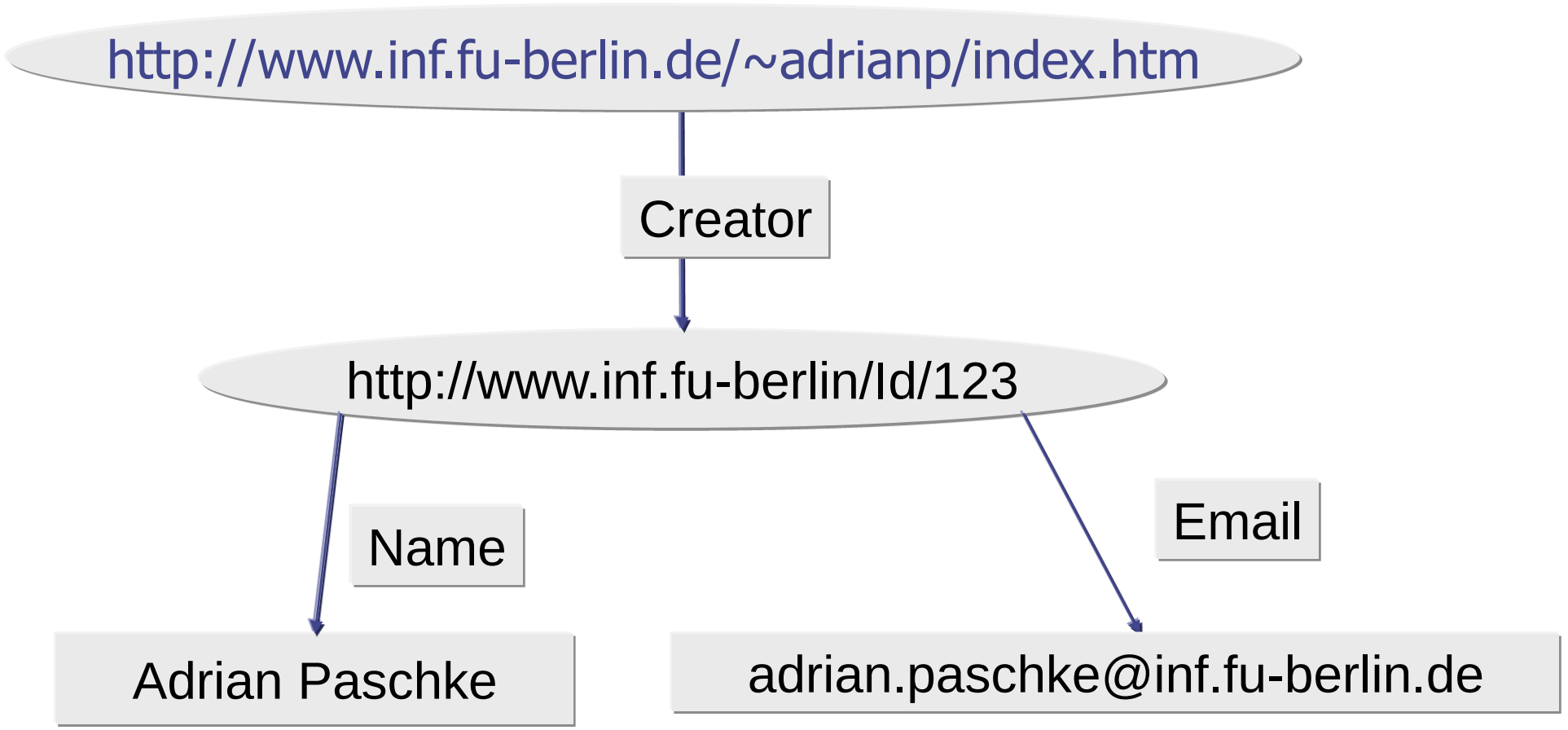
Extended RDF Diagram



RDF/XML Version

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

  <rdf:Description about="http://www.inf.fu-berlin.de/~adrianp/index.htm">
    <c:Creator>
      <rdf:Description>
        <c:Name>Adrian Paschke</c:Name>
        <c:Email>adrian.paschke@inf.fu-berlin.de</c:Email>
      </rdf:Description>
    </c:Creator>
  </rdf:Description>

</rdf:RDF>
```

Extended RDF Diagram



CSW
Corporate Semantic Web

RDF/XML-Version

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

  <rdf:Description about=" http://www.inf.fu-berlin.de/~adrianp/ ">
      <s:Creator rdf:resource="http:// www.inf.fu-berlin.de/Id/123 "/>
  </rdf:Description>

  <rdf:Description about=" http:// www.inf.fu-berlin.de/Id/123 ">
      <s:Name>Adrian Paschke</s:Name>
      <s:Email>adrian.paschke@inf.fu-berlin.de</s:Email>
  <rdf:Description>

</rdf:RDF>
```

Container in RDF

- Bag:
  - An unordered collection
- Sequence:
  - An ordered collection
- Alternative:
  - Unordered set of alternatives

Example Container Typ „Alternative“

```
<rdf:RDF>
  <rdf:Description about="http://x.org/packages/X11">
    <c:DistributionSite>
      <rdf:Alt>
        <rdf:li resource="ftp://ftp.x.org"/>
        <rdf:li resource="ftp://ftp.cs.purdue.edu"/>
        <rdf:li resource="ftp://ftp.eu.net"/>
      </rdf:Alt>
    </c:DistributionSite>
  </rdf:Description>
</rdf:RDF>
```

RDF Collections

- A limitation of these containers is that there is no way to **close** them
  - “these are **all** the members of the container”
- RDF provides support for describing groups containing **only** the specified members, in the form of RDF collections
  - **list** structure in the RDF graph
  - constructed using a predefined collection vocabulary: **rdf:List**, **rdf:first**, **rdf:rest** and **rdf:nil**

Reification

- In RDF it is possible to make statements about statements
  - **Adrian believes that Markus is the creator of http://www.corporate-semantic-web.de**
- Such statements can be used to describe belief or trust in other statements
- The solution is to assign a unique identifier to each statement
  - It can be used to refer to the statement

Reification (2)

- Introduce an auxiliary object (e.g. **belief1**)
- relate it to each of the 3 parts of the original statement through the properties **subject**, **predicate** and **object**
- In the preceding example
  - subject of **belief1** is **Markus**
  - predicate of **belief1** is creator
  - object of **belief1** is **http://www.corporate-semantic-web.de**

Data Types

- Data types are used in programming languages to allow interpretation
- In RDF, typed literals are used, if necessary

(“Markus”,

http://www.mydomain.org/age,

“27”^^http://www.w3.org/2001/XMLSchema#integer)

Data Types (2)

- ^^-notation indicates the type of a literal
- In practice, the most widely used data typing scheme will be the one by XML Schema
  - But the use of **any** externally defined data typing scheme is allowed in RDF documents
- XML Schema predefines a large range of data types
  - E.g. Booleans, integers, floating-point numbers, times, dates, etc.

Data Types

- The attribute rdf:datatype="&xsd:integer" is used to indicate the data type of the value of the age property

```
<rdf:Description rdf:about="949318">
    <uni:name>Markus</uni:name>
    <uni:title>Researcher</uni:title>
    <uni:age
    rdf:datatype="&xsd:integer">27<uni:age>
</rdf:Description>
```

RDF Metadata Languages

Some Examples

Dublin Core – Metadaten Vocabulary

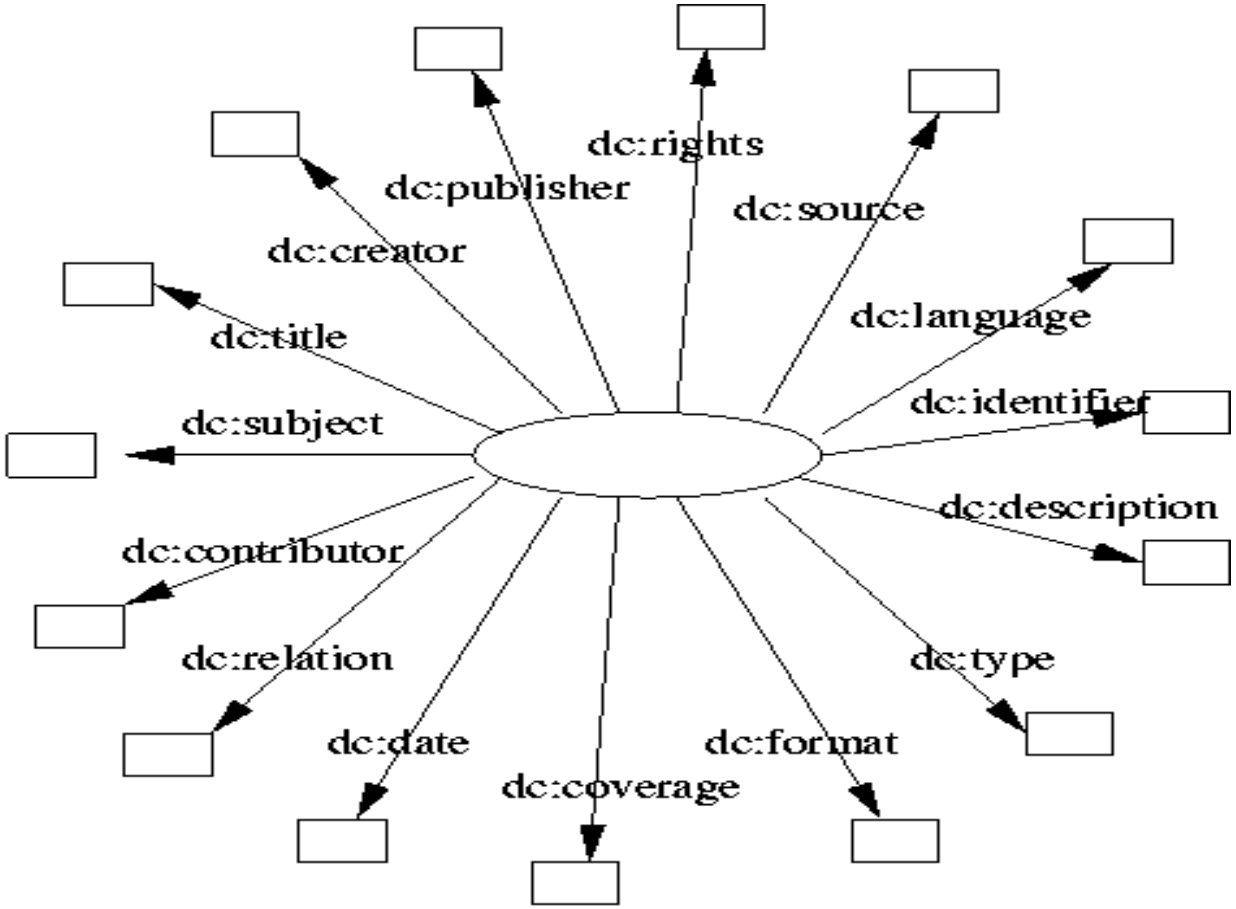
- Dublin Core Metadata Element Set can be represented in different syntax formats (e.g. RDF or HTML)
  - 15 Dublin Core core elements

- In HTML:

```
<HTML><HEAD>
<TITLE>IBIS Homepage</TITLE>
<META NAME=„DC.TITLE“ CONTENT=„RuleResponder“>
<META NAME=„DC.SUBJECT“ CONTENT=„rule inference
  services, decision support systems“>
<META NAME=„DC.CREATOR“ CONTENT=„A. Paschke“>
</HEAD>
…
```

Corporate Semantic Web

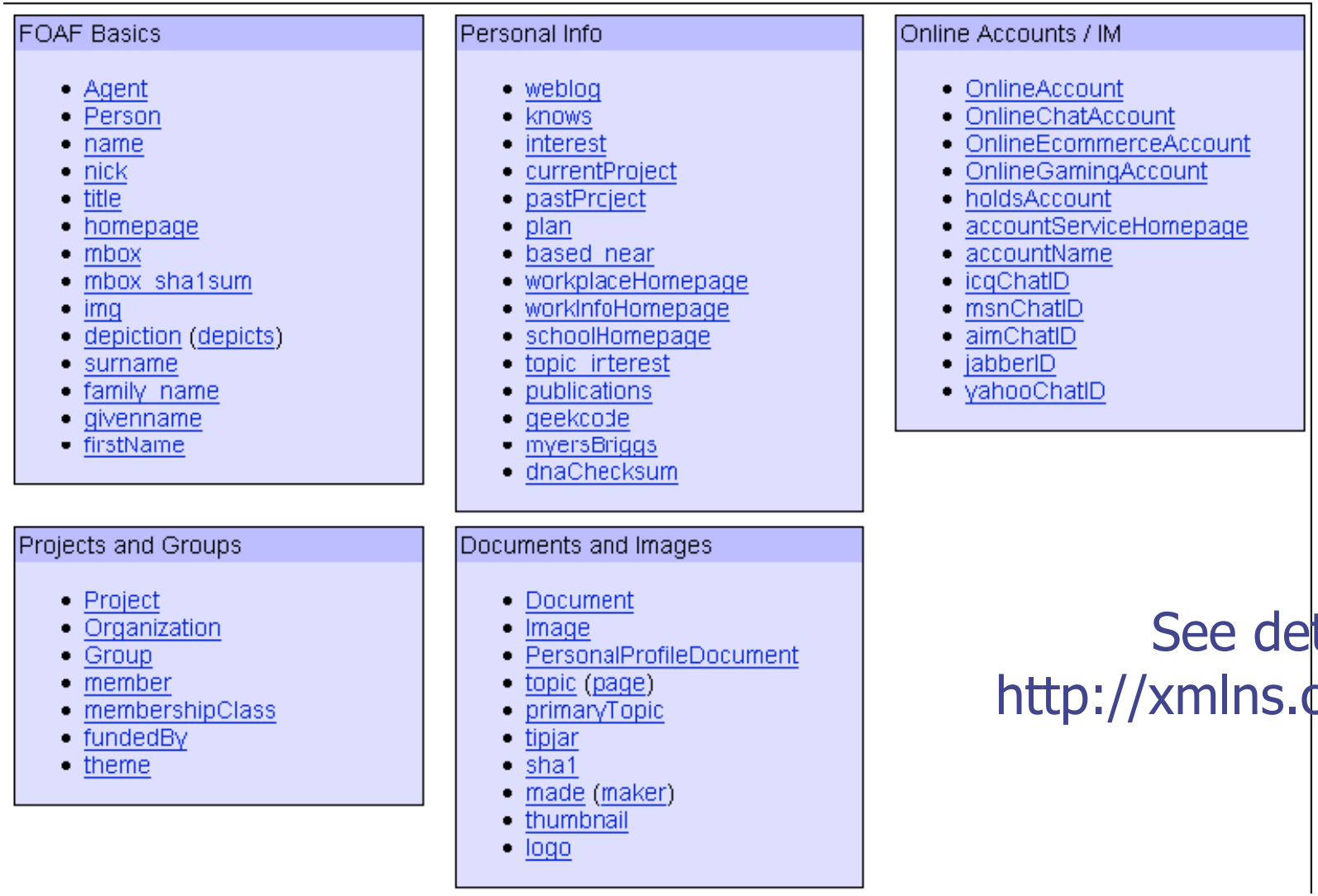
Dublin Core Elemente



Dublin Core in RDF

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
                xmlns:dc="http://purl.org/dublin_core/schema/">
  <rdf:Description rdf:about="responder.ruleml.org">
    <dc:creator>A. Paschke</dc:creator>
    <dc:title>Rule Responder</dc:title>
  </rdf:Description>
</rdf:RDF>
```

FOAF 0.1



See details at
http://xmlns.com/foaf/0.1/

CSW
Corporate Semantic Web

RDFa – RDF in HTML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
    "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    version="XHTML+RDFa 1.0" xml:lang="en">
  <head>
    <title>John's Home Page</title>
    <base href="http://example.org/john-d/" />
    <meta property="dc:creator" content="Jonathan Doe" />
  </head>
  <body>
    <h1>John's Home Page</h1>
    <p>My name is <span property="foaf:nick">John D</span> and I like
      <a href="http://www.neubauten.org/" rel="foaf:interest"
        xml:lang="de">Einstürzende Neubauten</a>.
    </p>
    <p>
      My <span rel="foaf:interest" resource="urn:ISBN:0752820907">favorite
      book</span> is the inspiring <span about="urn:ISBN:0752820907"><cite
      property="dc:title">Weaving the Web</cite> by
      <span property="dc:creator">Tim Berners-Lee</span></span>
    </p>
  </body>
</html>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://example.org/john-d/">
    <dc:creator xml:lang="en">Jonathan Doe</dc:creator>
    <foaf:nick xml:lang="en">John D</foaf:nick>
    <foaf:interest rdf:resource="http://www.neubauten.org/"/>
    <foaf:interest>
      <rdf:Description rdf:about="urn:ISBN:0752820907">
        <dc:creator xml:lang="en">Tim Berners-Lee</dc:creator>
        <dc:title xml:lang="en">Weaving the Web</dc:title>
      </rdf:Description>
    </foaf:interest>
  </rdf:Description>
</rdf:RDF>
```

SPARQL

A RDF Query Language

RDF Triple Stores

- A specialized **database for RDF triples**
- Supports a query language
  - **SPARQL** is the W3C recommendation
  - Might or might not do inferencing (e.g. ontologies)
  - Most query languages don't handle inserts
    - But e.g. SPARQL Update language
- Triple stores might be in memory or provide a persistent backend
  - Presistence provided by an underlying relational DBMS (e.g., mySQL) or a custom DB for efficiency.

SPARQL

- SPARQL (SPARQL Protocol and RDF Query Language)
  - “SQL for RDF”
  - A W3C standard Recommendation since 15 January 2008; SPARQL 1.1 since Nov 2012
- Basic concept “**Graph Pattern Matching**”
- Simple Protocol and RDF Query Language
  - Basic Graph Patterns (Conjunctive queries)
  - UNIONS
  - GRAPH Patterns
  - OPTIONAL Patterns
  - FILTERs

SPARQL Query Forms

- SELECT
  - Returns all, or a subset of, the variables bound in a query pattern match. Formats for the result set can be in XML or RDF/XML
- CONSTRUCT
  - Returns either an RDF graph that provides matches for all the query results or an RDF graph constructed by substituting variables in a set of triple patterns.
- DESCRIBE
  - Returns an RDF graph that describes the resources found.
- ASK
  - Returns whether a query pattern matches or not.

SPARQL SELECT

- SELECT:

```
SELECT Variables
FROM Dataset
WHERE Pattern
```

- Examples:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
  SELECT ?name
  WHERE ( ?x foaf:name ?name )
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
  SELECT *
  WHERE ( ?x foaf:name ?name )
```

SPARQL CONSTRUCT

- CONSTRUCT:

```
PREFIX: vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
CONSTRUCT * WHERE ( ?x vcard:FN ?name )

PREFIX foaf: <http://xmlns.com/foaf/0.1/> PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
CONSTRUCT ( ?x foaf:name ?name ) WHERE (?x vcard:FN ?name )
```

SPARQL DESCRIBE and ASK

- DESCRIBE:

```
DESCRIBE ?x
WHERE (?x ent:employeeId "1234")
```

- ASK:

```
PREFIX foaf:
  <http://xmlns.com/foaf/0.1/> ASK (?x
  foaf:mbox_sha1sum "ABCD1234")
```

Summary

- The Semantic Web is an extension of the Web with **semantic knowledge representation** formats which make the Web information understandable for machines
- **Resource Description Framework (RDF)** is a Web data information model to describe resources on the Web in a machine readable and interpretable way (abstract syntax maps to formal semantics of RDF)
  - RDF statements are subject-predicate-object triples consisting of resources, properties and values which can be literals or again resources
  - RDF can be represented as triple statements, as RDF/XML, as graph and can be stored in Triple Stores and e.g. embedded in HTML with RDFa
- **SPARQL** is a graph-based query language for RDF data
  - Select, Construct, Describe, Ask, (+ updates in SPARQL Update)

Questions

- What is the difference between syntax, semantics and pragmatics?
- What is the goal of the Semantic Web? What are the main building blocks of the Semantic Web?
- Name five W3C standards of the Semantic Web stack from 2007 and describe what they are used for?
- What is RDF? Describe the RDF data model. Name three different views on a RDF statement
- What is SPARQL? Name the four different query forms that SPARQL supports. Give an example of a SPARQL select query.

Web Ressourcen

- RDF Specification
  - http://www.w3.org/RDF/
- Ora Lassila; Introduction to RDF Metadata
  - http://www.w3c.org/TR/NOTE-rdf-simple-intro-971113.html
- SPARQL Specification
  - http://www.w3.org/2009/sparql/wiki/Main_Page