Introduction to the Semantic Web Technologies

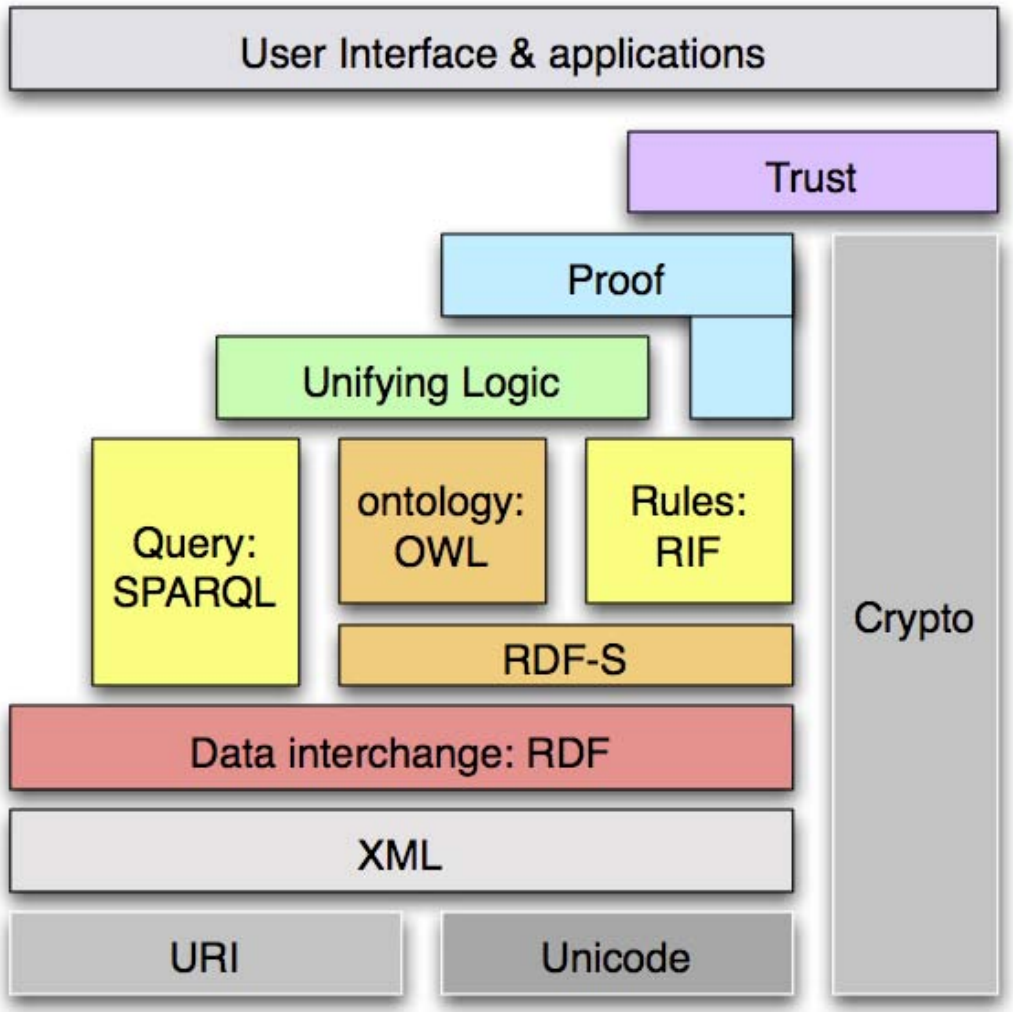
Konstantin Schekotihin

Alpen-Adria-Universität
Klagenfurt, Austria

Agenda

1. Basic concepts
   - Uniform Recourse Identifier
   - eXtensible Markup Language
2. RDF
   - Data exchange on the Web
   - Resource Description Framework
   - RDF Serialization
   - RDF Essentials
   - RDF Microformats

Architecture of the Semantic Web



URI: Uniform Resource Identifier

- URI = URL ∪ URN
  - URL (Uniform Resource Locator) identifies how and where a resource can be accessed

Example

`http://www.uni-klu.ac.at?query=value#fragment`

  - URN (Uniform Resource Name) persistent, location independent identifier

Example

`urn:issn:0004-3702` corresponds to ISSN number of "Artificial Intelligenge" book

- New W3C recommendations use IRI (Internationalized Resource Identifier) which is a generalized version of URI that is based on Unicode

XML: eXtensible Markup Language

- W3C standard for data exchange[1]
  - Applications can read, save and exchange data in XML format
  - XML Schema can be used to define vocabulary and allowed structures
  - XQuery / XPath to retrieve data from collections of XML documents
- XML extends HTML (XHTML)
  - HTML is used for presentation
  - XML for metadata
  - XHTML documents require strict well-formed syntax, whereas HTML parsers are flexible and can handle incorrect syntax

[1] http://www.w3.org/TR/xml11

XML Syntax

- XML begins with a declaration of used version and encoding

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

- Every XML document has only one root element
- XML elements:
  - data surrounded by a matching pair of tags
  - can contain text or another elements
- XML attributes are name/value pairs associated with corresponding XML elements

```
<book isbn13="9780132071482">
  <title>Artificial Intelligence</title>
  <subtitle>a modern approach</subtitle>
  The most popular AI textbook in the world.
</book>
```

Namespaces

- `xml:base=URI` defines a base namespace of an element that is different from the base URI of the document or parent elements. It is also used to shorten URIs in XML documents
- `xmlns:namespacePrefix=URI` the most common definition of namespaces for an XML subtree
- `<!ENTITY entity 'text'>` entity declarations of Document Type Definitions (DTD) can be used only once in an XML document in the DOCTYPE. These definitions affect the whole document:

```
<?xml version='1.0'?>
<!DOCTYPE foo [
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#"> ]>
  <foo name="XSD String Datatype" type="&xsd;string"/>
```

Data exchange on the Web

- Web applications from different vendors cannot use XML/XML Schema as universal information encoding format:
- The same information can be encoded using different schemas
  - Which schema is better?
- Merging two or more XML documents with the same information and different schemas is problematic
  - How should one match names of elements in one schema to the names in another?

XML can be used as a serialization language storing data in some universal format

Universal format for existing data

- Data is mostly stored in tabular form (databases)

MNr	Surname	Course	Note
12345	Müller	KE	1
12346	Schmid	LLP	2

- The Web story: Data is stored on different servers around the network
- Possible solutions:
  1. Store data row by row (common schema)
  2. Distribute columns (common entities/keys)
  3. Cell-oriented storing (requires both schema and entities)
- Distribution of cells allows to combine flexibility of the other two approaches:
  Servers a not limited neither to a schema nor to entities
- To represent a cell we need three values – triple

Triple representation

MNr	Surname	Course	Note
12345	Müller	KE	1
12346	Schmid	LLP	2

- Introduce some globally unique identifier for each row

Example

http://campus.uni-klu.ac.at/Student12345

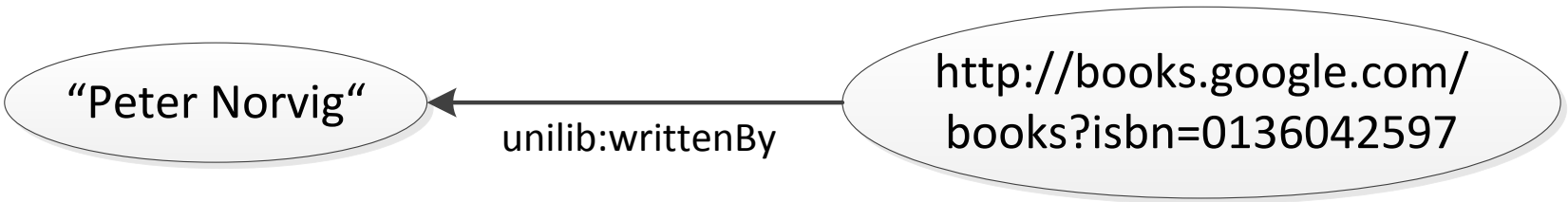
- Use column names as relation name
- Namespaces can simplify your life!

Example

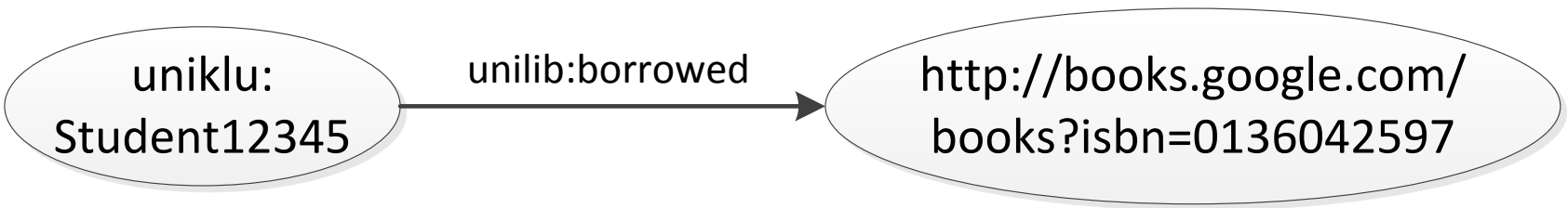
```
1 uniklu:Student12345 uniklu:MNr 12345
2 uniklu:Student12345 uniklu:Surname "Müller"
```

Data as a graph

- A set of triples can be efficiently represented as a graph



- Merge: Globally unique identifiers allow simple merging of graphs



- RDF graph is described as a set of triples also called statements

<Subject, Predicate, Object>

- Subjects and objects can be
  - URIs to reference resources

Example

`http://books.google.com/books?as_isbn=9780132071482` references a book using its ISBN number, which is a unique identifier

  - anonymous or blank nodes. This allows to model incomplete information, i.e. describe some objects which name is unknown or irrelevant.

- Objects can also be Literals to save data values "Peter Norvig" is a literal of the type `xsd:string`
- Predicates are always expressed using URIs

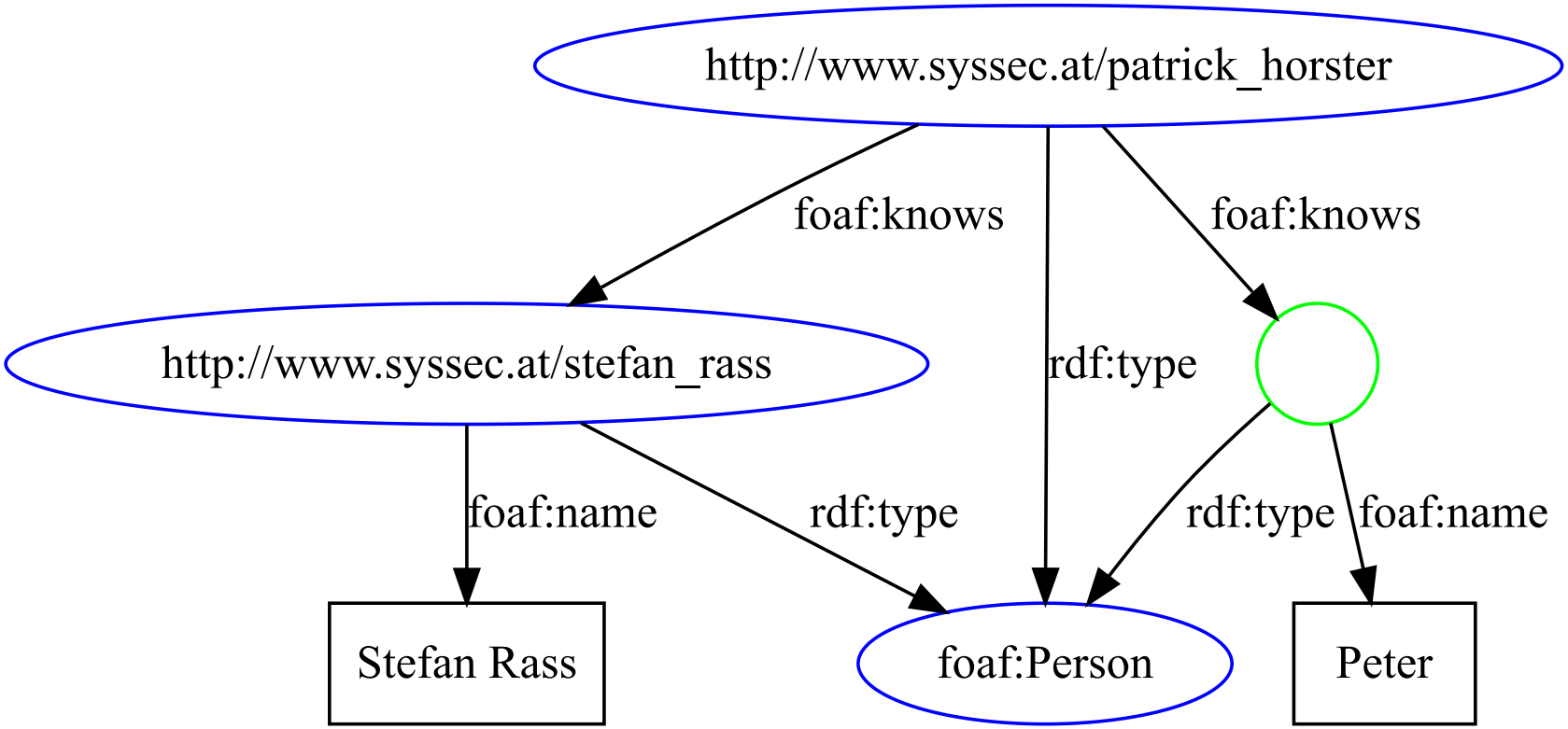
RDF/XML Syntax

RDF/XML syntax is the main serialization method for RDF[2]

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:foaf = "http://xmlns.com/foaf/0.1/">
<!-- Subject -->
<foaf:Person rdf:about="http://www.syssec.at/patrick_horster">
   <foaf:knows> <!-- Predicate -->
      <foaf:Person rdf:about="http://www.syssec.at/stefan_rass">
         <!-- Object -->
         <foaf:name>Stefan Rass</foaf:name>
      </foaf:Person>
   </foaf:knows> <foaf:knows>
      <foaf:Person>
         <foaf:name>Peter</foaf:name>
      </foaf:Person>
   </foaf:knows>
</foaf:Person>
</rdf:RDF>
```

[2] http://www.w3.org/TR/rdf-syntax-grammar

Graph Representation[3]



[3] Generated using EasyRDF `http://www.easyrdf.org/converter`

RDF/XML: Long Variant

```
<?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/">

<rdf:Description rdf:about="http://www.syssec.at/patric_horster">
   <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
    <foaf:knows rdf:resource="http://www.syssec.at/stefan_rass"/>
    <foaf:knows rdf:nodeID="node18vq12l5ux98655"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.syssec.at/stefan_rass">
   <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
   <foaf:name>Stefan Rass</foaf:name>
</rdf:Description>

<rdf:Description rdf:nodeID="node18vq12l5ux98655">
   <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
   <foaf:name>Peter</foaf:name>
</rdf:Description>
</rdf:RDF>
```

RDF Turtle Syntax

Turtle[4] syntax is adopted to make RDF documents human-readable and is also used as the basis for the syntax of SPARQL

```
@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf:<http://xmlns.com/foaf/0.1/> .
@prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#> .
@prefix sec:<http://www.syssec.at/> .

foaf:name rdf:type rdf:Property .
sec:stefan_rass rdf:type  foaf:Person .
sec:stefan_rass foaf:name "Stefan Rass" .
sec:patric_horster rdf:type foaf:Person;
                   foaf:knows sec:stefan_rass,
                              [ a foaf:Person;
                                foaf:name "Peter" ] .
```

[4] http://www.w3.org/TeamSubmission/turtle
Validator http://any23.org

RDF Essentials I

- All XML Schema datatypes can be used in RDF
- Values of literals are interpreted accordingly to their datatypes, e.g. "123"^^xsd:integer ("00123"^^xsd:decimal)
- rdf:XMLLiteral is the only RDF datatype that allows to include XML fragments

```
@prefix ex: <http://example.org/stuff/1.0/> .
@prefix xmlns: <http://www.w3.org/2000/xmlns/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
ex:item01 ex:prop
   "<a:Box required=\"true\"
    xmlns:a=\"http://example.org/a#\">
       <a:widget size=\"10\" />
       <a:grommit id=\"23\" />
    </a:Box>
    "^^rdf:XMLLiteral .
```

RDF Essentials II

- RDF allows to define three types of open containers:
  - `rdf:Seq` ordered list,
  - `rdf:Bag` unordered set,
  - `rdf:Alt` set of alternatives.
- In Turtle the ordered list can be defined as follows

```
@prefix ex: <http://example.org/stuff/1.0/> .
ex:Nikon_D4 ex:Sensitivity (ex:100 ex:200 ex:400) .
```

Reification

- How to model propositions about propositions: “Wikipedia states that Guinness is brewed by St. James Gate Brewery”
- We can use the RDF reification vocabulary:
  - `rdf:Statement` is intended to represent the class of RDF statements.
  - `rdf:subject`, `rdf:predicate` and `rdf:object` are used to state the subject, predicate and object of a statement.

```
@prefix ex: <http://example.org/stuff/1.0/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
ex:Wikipedia ex:states ex:statement1 .
ex:statement1 rdf:subject ex:Guiness;
              rdf:predicate ex:brewed;
              rdf:object ex:St_James_Gate_Brewery.
```

RDFa: RDF-in-attributes I

- RDF triples in XHTML attributes are indexed be search engines to improve ranking of documents in search results
- W3C Recommendation 2012 [5] (version 1.1)
- Easy to integrate with existing HTML based applications

Example taken from http://www.uni-klu.ac.at/english

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de"
    lang="de">
...
<h2>Welcome to the English website of the
    Alpen-Adria-Universitaet Klagenfurt!</h2>
...
</html>
```

RDFa: RDF-in-attributes II

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de"
     lang="de"
    xmlns:uni="http://example.org/univerity"
>
...
<div about="http://dbpedia.org/page/University_of_Klagenfurt ">
    <h2 xmlns:lang="http://example.org/languages">
     Welcome to the <span property="lang:resource">English</span>
     website of the
     <span property="uni:name">Alpen-Adria-Universitaet
     Klagenfurt</span>!
    </h2>
</div>
...
</html>
```

[5] http://www.w3.org/TR/rdfa-syntax