



University of Dublin
Trinity College



Resource Description Framework (RDF)

Melike Şah Direkoğlu

*Adapted from course notes of
Rob Brennan, TCD, Declan O'Sullivan, TCD, Simon Dobson, UCD and Myungjin Lee, LIST*

Problem of XML

Park John Smith has the phone number (+82)-10-3099-9183.

```
<person>
  <name>John Smith</name>
  <tel>(+82)-10-3099-9183</tel>
</person>
```

```
<person name="John Smith">
  <tel>(+82)-10-3099-9183</tel>
</person>
```

```
<person name="John Smith" tel="(+82)-10-3099-9183" />
```

→ We need a method to represent data on abstract level.

Representing Knowledge

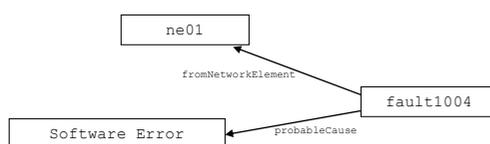
There are a number of options

- As *objects*, using the well-accepted techniques of object-oriented analysis and design to capture a model
- As *clauses*, going back to the early days of AI and Lisp
- As *XML*, using the industry-standard structured mark-up language
- As *graphs*, making use of the things we know about graph theory
- As some combination of these

We are looking for: extensibility, ease of use, ease of querying

Which would *you* choose?

Graphs

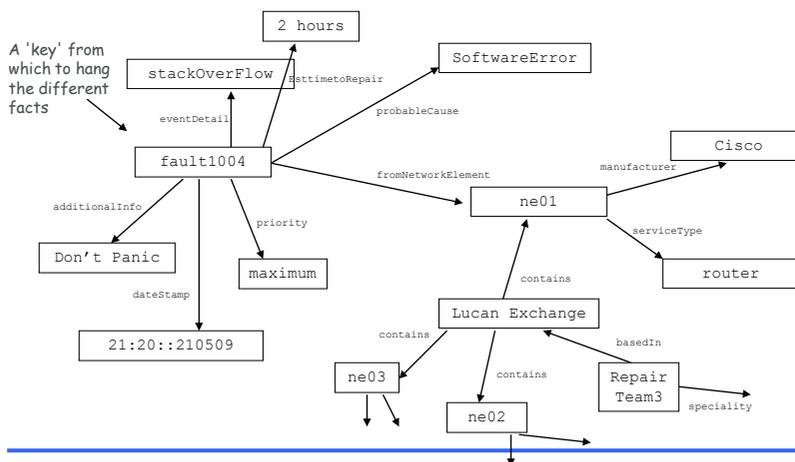


We can use the nodes of a graph for facts and the arcs as (binary) relationships between them

- Arcs are typically called **predicates** or **relationships** in this view
- The set of arcs intersecting a node tells us the information we know about that fact or entity

Graphs as knowledge – 1

How do we use graphs to represent knowledge?



Graphs as knowledge – 2

Things to note

- Scaling – the same graph can represent a load of different knowledge simultaneously
 - Agreement – need to know what the various predicates 'mean'
 - Structure – you need to know what nodes are related by a predicate
 - Plurality – the same relationship may appear several times
 - Symmetry – the same predicates can be used for common information, despite minor changes
 - Asymmetry – relationships are inherently directed, which sometimes makes things awkward
- ...and this can get very tricky
- ...and this can be difficult to keep straight
- For example both NetworkElements and Faults might have estimateTimeToRepair
- So a knowledge graph is inherently directed

Two ways to view a graph

As nodes and arcs

- Nodes store facts, edges store relationships between them



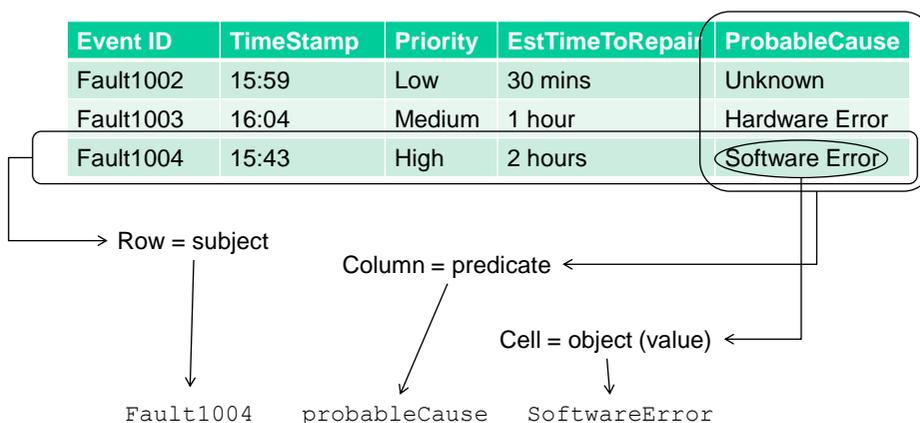
As triples

- A three-place relationship of 'subject, predicate, object'
- The node and edge structure is induced by the triples – each triple defines an edge, the different subjects/objects are the population of nodes, one node per individual string

Fault1004 probableCause SoftwareError

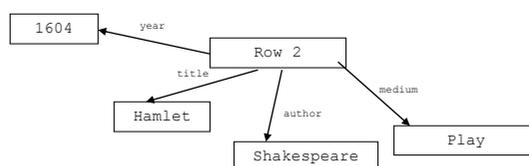
The clause form we saw earlier is essentially this triple form but using the order 'predicate subject object'

Relationship to Relational Data?



Relationship to Relational Data?

ID	Title	Author	Medium	Year
1	As You Like It	Shakespeare	Play	1599
2	Hamlet	Shakespeare	Play	1604
3	Hero and Leander	Marlowe	Poem	1593



Graph based approach - 1

The promise

- natural distribution
- easy merging
- naturally extendible
- easy publication and consumption
- easy querying (?)

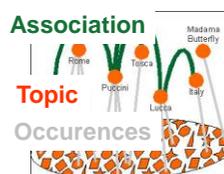
Which one?

Requirements

- easily processed
- standards based
- Easily published/consumed

The Contenders

- XML Topic Maps (XTM)
 - ISO Standard
 - Small vocabulary
 - Simple TAO approach ("everything is a topic")
 - Easy navigation and querying
 - No standard reasoners
- Resource Description Framework (RDF)
 - W3C Standard
 - Small vocabulary
 - More tricky to model
 - Navigation good, Querying tricky
 - Reasoning capable



RDF,RDFS

© Declan O'Sullivan

11

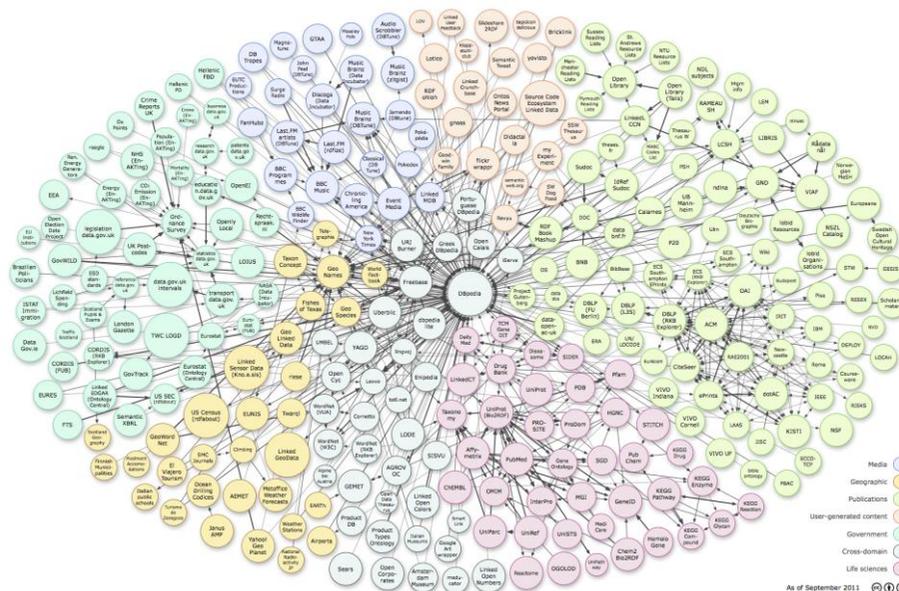
The Long Road for RDF

- Fits and Starts
 - Original spec 1999
 - Started to have more traction 2004
 - Adopted as baseline in Semantic Web community
- Linked Data Movement
 - Tim Berners Lee driven
 - Treat schemas as vocabularies
 - Reuse existing schemas
 - foaf, sioc, dc
 - <http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData/>

RDF,RDFS

© Declan O'Sullivan

12



Resource Description Framework (RDF)

RDF is a **graph-based data model** that allows us to identify things, classes of things and labelled parts of things in a standard way.

Standardised by the World Wide Web Consortium (W3C)

RDF has different syntaxes; XML (standard), N3, JSON-LD, Terse, etc.

RDF isn't a knowledge standard *per se*: it's a way of **defining** knowledge standards in a way that maximises the potential for re-use across the web

- A way of defining knowledge graphs
- **No standard predicates**
- Tool support – editors, parsers, displays, .

Lassila and Swick, Resource Description Framework model and specification. W3C report, 1999

Thus RDF is

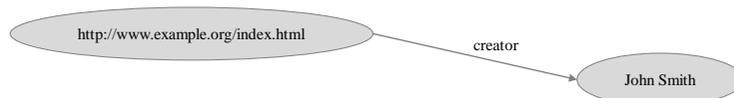
used as a **general method** for **conceptual description or modeling** of information that is implemented in **web resources**, using a variety of syntax notations and data serialization formats

based on the idea that the **things** being described have **properties which have values**, and that resources can be described **by making statements (triples)**

15

Terminology for the Various Parts of Statements

http://www.example.org/index.html has a **creator** whose value is **John Smith**



- the *subject* is the URL `http://www.example.org/index.html`
 - the *predicate* is the word "creator"
 - the *object* is the phrase "John Smith"
-

16

What does RDF give us?

RDF is intended to address some of the issues we've identified in representing knowledge

- Extensible – easy to add new information
- Simple – XML is (pretty) simple to manipulate
- Standard – defined by a standards body
- Scalable – used on the widest (Internet) scale

The way it accomplishes this is what we'll look at next

Note that this isn't a complete list of issues - no mention of query complexity or how to actually represent large models

Basic structure – triples

RDF represent knowledge using a triple structure

- Subject
- Predicate
- Object

Remember, a triple structure is one way of viewing a graph, so RDF essentially defines a knowledge graph

Knowledge is built up as a collection of these triples, contained within an XML file or other serialization format.

Two Things for Processing of RDF by Machines

a system of machine-processable identifiers for identifying a subject, predicate, or object in a statement without any possibility of confusion with a similar-looking identifier that might be used by someone else on the Web.

→ **Use URI (Uniform Resource Identifier)**

a machine-processable language for representing these statements and exchanging them between machines.

→ **Use XML syntax**

19

Identifying “things” (resources) with URI

RDF re-uses the URI as a global namespace of identifiers for things.

- Unique across entire WWW
- URIs can contain URLs that can be de-referenced (resolved) to find out more info about the “thing”
- If two resources use the same URI => they are the same thing

Back to triples:

- An RDF subject is always a resource => always a URI
- An RDF object can be a resource or a literal value
- What about predicates? (always a URI)

Namespaces and URIs

Namespaces use URIs, and URIs can be made unique

- If I want to define a new structure I just define a namespace and assign it a unique URI
- If I own a domain I can give it any name I want under my domain name, secure in the knowledge that no-one else will (should!) use it

So a set of predicates, subjects, objects defined using a namespace can be uniquely differentiated from any other set of predicates across the entire web

- Cheap, decentralised model

Common sets of predicates may be given well-known names and URIs

Some common RDF namespaces

RDF: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
RDFS: <http://www.w3.org/2000/01/rdf-schema#>
OWL: <http://www.w3.org/2002/07/owl#>
XML schema <http://www.w3.org/2001/XMLSchema#>
FOAF: <http://xmlns.com/foaf/0.1/>
dc: <http://purl.org/dc/elements/1.1/>
dcterms: <http://purl.org/dc/terms/>

Two types of namespaces: slash(/) and hash(#)

- Hash implies a single file
 - Use for small, relatively static vocabularies
- Slash implies a set of files or dynamic generation (RESTful)
 - Use for large, dynamic vocabs

Predicate and URI Example

The set of predicates is only defined informally

Moreover pre-supposes a human language and some common understanding

- `is_a` makes sense to an English speaker, but isn't so good for a Swahili-speaker

We need to get broad agreement on what the various symbols mean

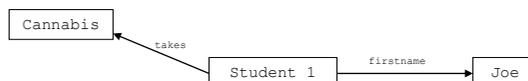
- Meaningless to a computer anyway, of course...
- ...but we have to make sure we use them consistently
- ...and on an Internet scale

Predicate and URI Example

Consider the word 'takes'



Could also mean it like this:



The word 'takes' – and indeed *any* word – is open to mis-interpretation

- Not precisely defined without a context (sic)
- ...and we need something better for machines to work with

Thus use URIs to uniquely identify the meaning of "takes"

Use the XML Syntax

To represent RDF statements in a machine-processable way, RDF uses the Extensible Markup Language.

RDF/XML

- for use in representing RDF information, and for exchanging it between machines

25



University of Dublin
Trinity College



RDF Statements

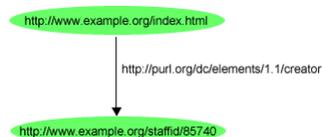
Statements

English Statement

- <http://www.example.org/index.html> has a **creator** whose value is **John Smith**

RDF Statement

- a subject <http://www.example.org/index.html>
- a predicate <http://purl.org/dc/elements/1.1/creator>
- and an object <http://www.example.org/staffid/85740>



RDF Graph Model

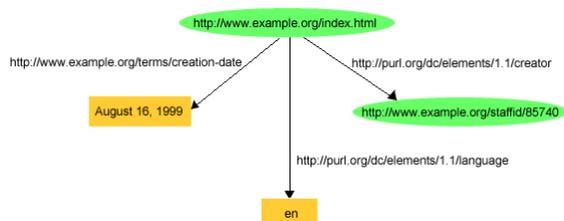
- a node for the subject
- a node for the object
- an arc for the predicate, directed from the subject node to the object node

27

Groups of Statements

<http://www.example.org/index.html> has a **creation-date** whose value is **August 16, 1999**

<http://www.example.org/index.html> has a **language** whose value is **English**

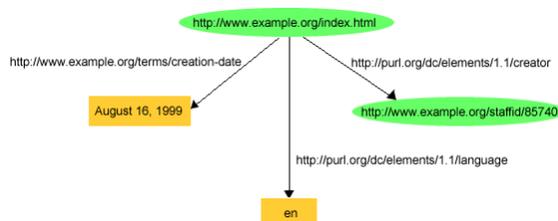


28

Triple

an alternative way of writing down the statements
written as a simple triple of subject, predicate, and
object, in that order

```
<http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/creator> <http://www.example.org/staffid/85740> .
<http://www.example.org/index.html> <http://www.example.org/terms/creation-date> "August 16, 1999" .
<http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/language> "en" .
```



29

Shorthand Way of Writing Triples

QName (XML Qualified Name)

- a valid identifier for elements and attributes
- a *prefix* that has been assigned to a namespace URI, followed by a colon, and then a *local name*

For Example,

- prefix `dc:`, namespace URI: `http://purl.org/dc/elements/1.1/`
- prefix `ex:`, namespace URI: `http://www.example.org/`
- prefix `exterms:`, namespace URI: `http://www.example.org/terms/`
- prefix `exstaff:`, namespace URI: `http://www.example.org/staffid/`

```
ex:index.html    dc:creator      exstaff:85740 .
ex:index.html    exterms:creation-date  "August 16, 1999" .
ex:index.html    dc:language     "en" .
```

30

URIref and Literal

URIref

- to identify the things
- shown as ellipses
- not only the subject of the original statement, but also the predicate and object

Literal

- constant values represented by character strings
 - shown as boxes
 - Literals may not be used as subjects or predicates in RDF statements. **Only as objects!**
-

31

Literal

Plain Literal

- a string combined with an optional language tag

Types Literal

- a string combined with a datatype URI
- based on XML Schema datatypes

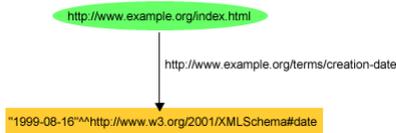


32

XML Schema Datatypes

Namespace of XML Schema

- <http://www.w3.org/2001/XMLSchema>



Datatypes

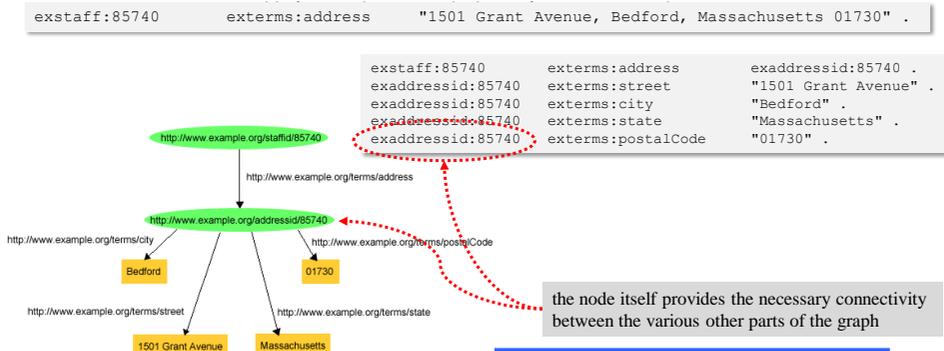
Simple Type	Examples
string	Confirm this is electric
integer	...-1, 0, 1, ...
long	-9223372036854775808, ... -1, 0, 1, ... 9223372036854775807
float	-INF, -1E4, -0, 0, 12.78E-2, 12, INF, NaN
date	1999-05-31
time	13:20:00.000, 13:20:00.000-05:00
boolean	true, false, 1, 0

33

Blank Nodes

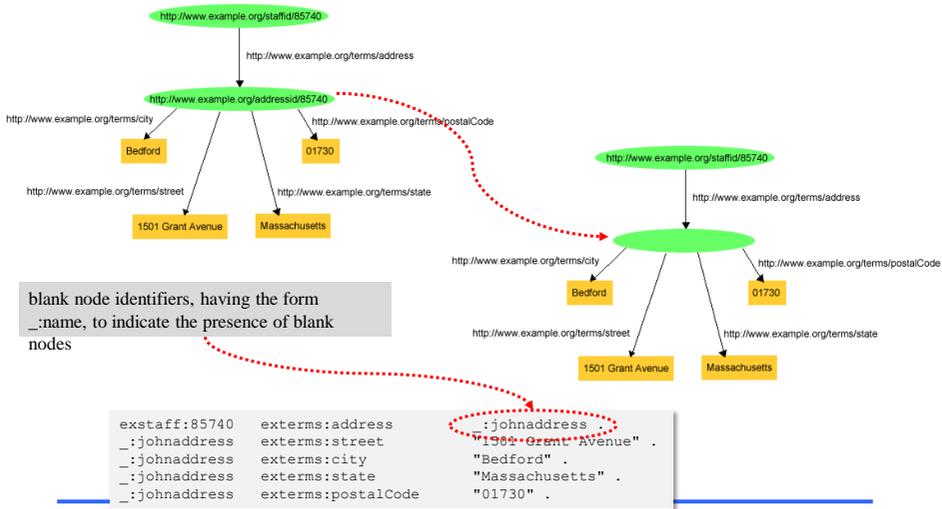
a way to represent multi-valued relationships

Why we need blank nodes?



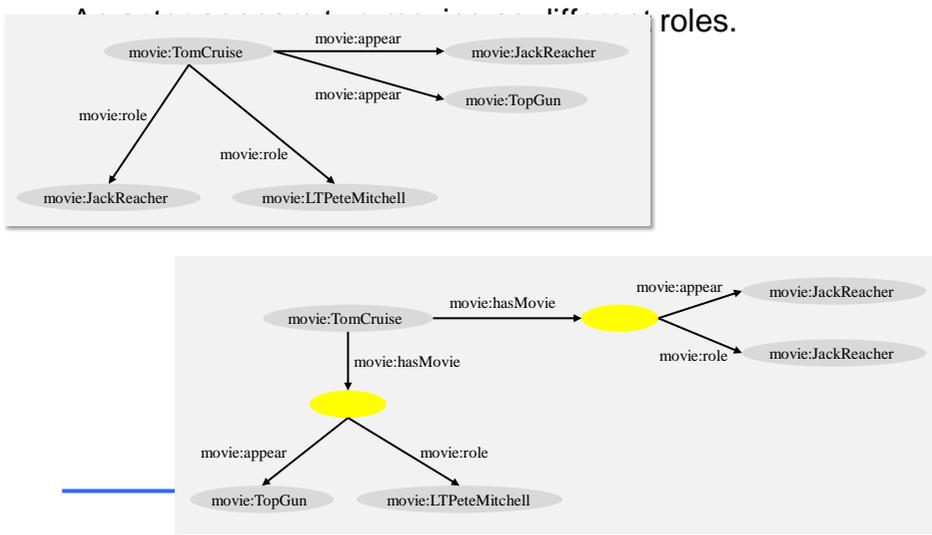
34

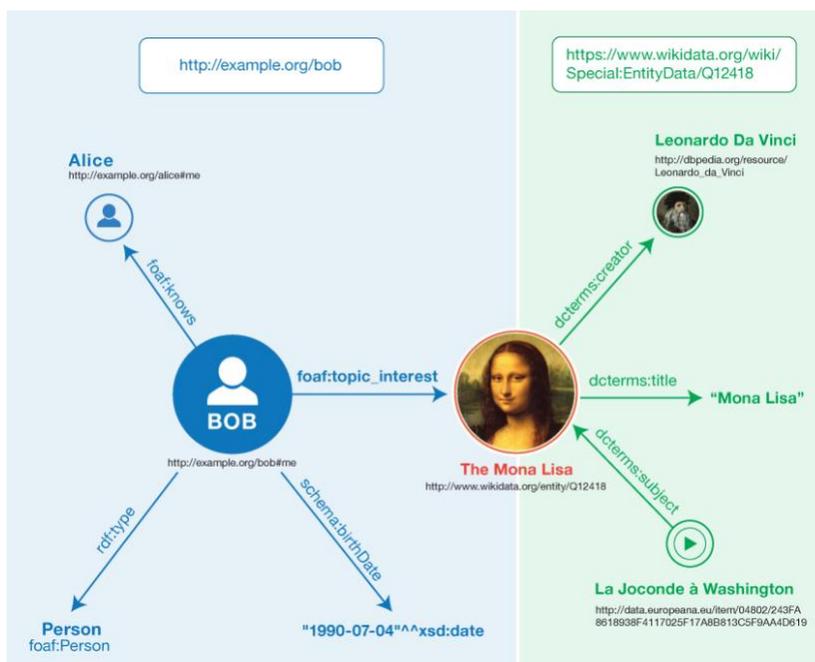
Blank Nodes



35

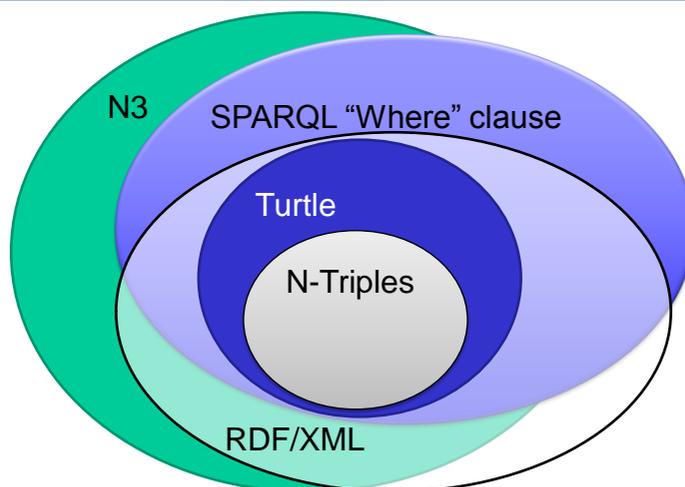
Example of Blank Nodes





SERIALISING RDF (SYNTAX)

Many solutions!



RDF,RDFS

© Declan O'Sullivan

39

RDF/XML

RDF != XML

RDF/XML is the only standardised serialisation of RDF

Most common format supported by tools

Uses a tree (XML) to represent a graph

- Trees have a root, graphs do not

RDF,RDFS

© Declan O'Sullivan

40

RDF/XML

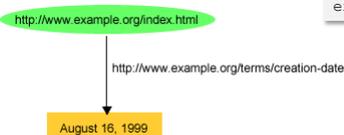
RDF/XML

- an XML syntax for writing down and exchanging RDF graphs
- the normative syntax for writing RDF

English Statement

`http://www.example.org/index.html` has a `creation-date` whose value is `August 16, 1999`

RDF Graph



Triple

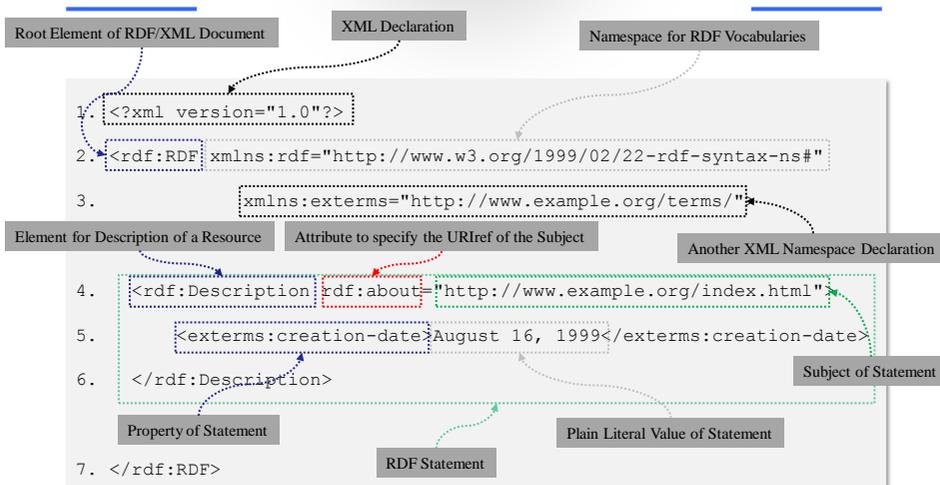
`ex:index.html` `exterm:creation-date` "August 16, 1999" .

RDF/XML

```

1. <?xml version="1.0"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.     xmlns:exterm="http://www.example.org/terms/">
4.   <rdf:Description rdf:about="http://www.example.org/index.html">
5.     <exterm:creation-date>August 16, 1999</exterm:creation-date>
6.   </rdf:Description>
7. </rdf:RDF>
    
```

RDF/XML Document



RDF/XML for Two Statements

Triples

```
ex:index.html    exterm:s:creation-date    "August 16, 1999" .
ex:index.html    dc:language                    "en" .
```

RDF/XML

```

1. <?xml version="1.0"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.     xmlns:dc="http://purl.org/dc/elements/1.1/"
4.     xmlns:exterm:s="http://www.example.org/terms/">
5.     <rdf:Description rdf:about="http://www.example.org/index.html">
6.         <exterm:s:creation-date>August 16, 1999</exterm:s:creation-date>
7.     </rdf:Description>
8.     <rdf:Description rdf:about="http://www.example.org/index.html">
9.         <dc:language>en</dc:language>
10.    </rdf:Description>
11. </rdf:RDF>

```

Abbreviating Multiple Properties

Triples

```
ex:index.html    dc:creator            exstaff:85740 .
ex:index.html    exterm:s:creation-date    "August 16, 1999" .
ex:index.html    dc:language                "en" .
```

RDF Graph

RDF/XML

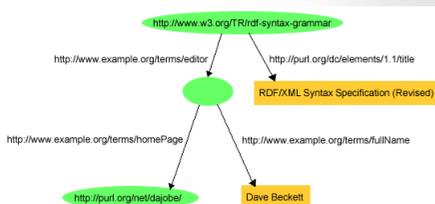
```

1. <?xml version="1.0"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.     xmlns:dc="http://purl.org/dc/elements/1.1/"
4.     xmlns:exterm:s="http://www.example.org/terms/">
5.     <rdf:Description rdf:about="http://www.example.org/index.html">
6.         <exterm:s:creation-date>August 16, 1999</exterm:s:creation-date>
7.         <dc:language>en</dc:language>
8.         <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
9.     </rdf:Description>
10. </rdf:RDF>

```

Attribute `rdf:resource` indicates that the property element's value is another resource

RDF/XML Describing a Blank Node



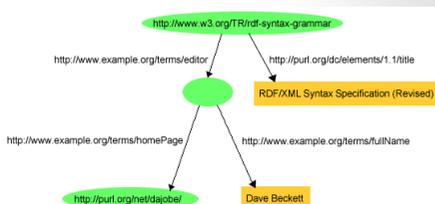
```

1. <?xml version="1.0"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.     xmlns:dc="http://purl.org/dc/elements/1.1/"
4.     xmlns:exterm="http://example.org/stuff/1.0/">
5.
6.     <rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
7.         <dc:title>RDF/XML Syntax Specification (Revised)</dc:title>
8.         <exterm:editor rdf:nodeID="abc"/>
9.     </rdf:Description>
10.
11.     <rdf:Description rdf:nodeID="abc">
12.         <exterm:fullName>Dave Beckett</exterm:fullName>
13.         <exterm:homePage rdf:resource="http://purl.org/net/dajobe/">
14.     </rdf:Description>
15. </rdf:RDF>

```

Attribute to declare and refer a blank node

Anonymous Blank Nodes



Attribute to declare Anonymous blank node

```

1. <?xml version="1.0"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.     xmlns:dc="http://purl.org/dc/elements/1.1/"
4.     xmlns:exterm="http://example.org/stuff/1.0/">
5.
6.     <rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
7.         <dc:title>RDF/XML Syntax Specification (Revised)</dc:title>
8.         <exterm:editor rdf:parseType="Resource">
9.             <exterm:fullName>Dave Beckett</exterm:fullName>
10.            <exterm:homePage rdf:resource="http://purl.org/net/dajobe/">
11.        </exterm:editor>
12.    </rdf:Description>
13. </rdf:RDF>

```

RDF/XML Using a Typed Literal

```
ex:index.html  exterts:creation-date  "1999-08-16"^^xsd:date .
```

Attribute to specify the datatype

```
1. <?xml version="1.0"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.     xmlns:exterts="http://www.example.org/terms/"
4.     <rdf:Description rdf:about="http://www.example.org/index.html">
5.         <exterts:creation-date rdf:datatype=
6.             "http://www.w3.org/2001/XMLSchema#date">1999-08-16
7.     </rdf:Description>
8. </rdf:RDF>
```

Defining Base URI

XML Base

- facility for defining base URIs for parts of XML documents
- to specify a base URI other than the base URI of the document or external entity

```
1. <?xml version="1.0"?>
2. <!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
3. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4.     xmlns:exterts="http://www.example.com/terms/"
5.     xml:base="http://www.example.com/2002/04/products">
6.     <rdf:Description rdf:about="#item10245">
7.         <exterts:model rdf:datatype="&xsd:string">Overnighter</exterts:model>
8.         <exterts:sleeps rdf:datatype="&xsd:integer">2</exterts:sleeps>
9.     </rdf:Description>
10. </rdf:RDF>
```



rdf:ID Attribute

rdf:ID Attribute

- to specify a *fragment identifier*, given by the value of the `rdf:ID` attribute
- interpreted relative to a *base URI* appending the character "#"

```

1. <?xml version="1.0"?>
2. <!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">>
3. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4.     xmlns:exterm="http://www.example.com/terms/"
5.     xml:base="http://www.example.com/2002/04/products">
6.
7.     <rdf:Description rdf:ID="item10245">
8.         <exterm:model rdf:datatype="xsd:string">Overnighter</exterm:model>
9.         <exterm:sleeps rdf:datatype="xsd:integer">2</exterm:sleeps>
10.    </rdf:Description>
11. </rdf:RDF>

```



49

[Myungjin Lee]

rdf:ID and rdf:about

The two forms are essentially synonyms: the full *URIref* formed by RDF/XML is the same in either case.

Using `rdf:ID` provides an additional check when assigning a set of distinct names.

- A given value of the `rdf:ID` attribute can only appear once relative to the same *base URI*.

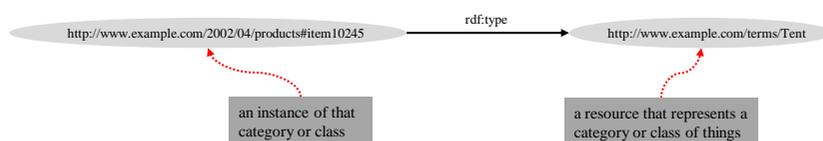
50

Typed Node

Typed Node

- the **resources described as instances** of specific types or classes
- being classified into different *kinds* or *categories*
- by providing a **predefined property, `rdf:type`**

```
1. <rdf:Description rdf:ID="item10245">
2.   <rdf:type rdf:resource="http://www.example.com/terms/Tent"/>
3. </rdf:Description>
```



51

Abbreviation for Describing Typed Nodes

How to describe abbreviation for typed nodes

- the `rdf:type` property and its value are removed
- the `rdf:Description` element is replaced by an element whose name is the *QName* corresponding to the value of the removed `rdf:type` property

```
1. <rdf:Description rdf:ID="item10245">
2.   <rdf:type rdf:resource="http://www.example.com/terms/Tent"/>
3. </rdf:Description>
```

replaced

```
1. <exterms:Tent rdf:ID="item10245" />
```



52

Structured Values

```
exproduct:item10245    exterm:weight    "2.4"^^xsd:decimal .
```

This is the decimal value of weight property using typed literal.

rdf:value Property

- to **describe the main value** (if there is one) of a structured value

How to add an indication of the unit of measure

```
exproduct:item10245    exterm:weight    _:weight10245 .
_:weight10245         rdf:value          "2.4"^^xsd:decimal .
_:weight10245         exterm:units     exunits:kilograms .
```

53

RDF/XML for Structured Value

```
exproduct:item10245    exterm:weight    _:weight10245 .
_:weight10245         rdf:value          "2.4"^^xsd:decimal .
_:weight10245         exterm:units     exunits:kilograms .
```

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:exterm="http://www.example.org/terms/">
  <rdf:Description rdf:about="http://www.example.com/2002/04/products#item10245">
    <exterm:weight rdf:parseType="Resource">
      <rdf:value rdf:datatype="&xsd;decimal">2.4</rdf:value>
      <exterm:units rdf:resource="http://www.example.org/units/kilograms"/>
    </exterm:weight>
  </rdf:Description>
</rdf:RDF>
```

rdf:parseType="Resource" attribute

- to indicate that the contents of an element are to be **interpreted as the description of a new (blank node) resource**, without a nested `rdf:Description` element

54

XML Literal

`rdf:parseType="Literal"` attribute

- to indicate that the contents of the element are to be **interpreted as an XML fragment**

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xml:base="http://www.example.com/books">

  <rdf:Description rdf:ID="book12345">
    <dc:title rdf:parseType="Literal">
      <span xml:lang="en">
        The <em>&lt;br /&gt;</em> Element Considered Harmful.
      </span>
    </dc:title>
  </rdf:Description>

</rdf:RDF>
```

55



University of Dublin
Trinity College



Turtle

Turtle – Terse RDF Triple Language

More human friendly/readable syntax

- Not XML based (just text)
- Does not have to represent a graph as a tree!

The same fact as before:

```
@prefix ff: <http://www.fame.ie/ontologies/fame-faults#>
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

ff:fault1004 ff:additionalInfo "OK panic now" .
```

Namespace declarations

Triple ends with a period

Subject, predicate, object separated by a whitespace all on one line

Turtle shortcuts

Multiple statements about the same subject:

```
@prefix ff: <http://www.fame.ie/ontologies/fame-faults#>
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

ff:fault1004 ff:additionalInfo "OK panic now" ;
             ff:additionalInfo "This is really bad" ;
             ff:priority         ff:high .
```

Semicolon indicates that next 2 elements refer to same subject.

Set of triples ends with a period

Multiple statements with the same subject and predicate

```
@prefix ff: <http://www.fame.ie/ontologies/fame-faults#>
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

ff:fault1004 ff:additionalInfo "OK panic now", "This is really bad" .
```

Comma indicates that next element refers to same subject with the same predicate.

Set of triples ends with a period



University of Dublin
Trinity College



RDFa

Concept

Add structured (RDF) meta-data to XHTML web-pages

Why?

- To Bridge web of data (machine-readable i.e. Semantic Web) and web of documents (human-readable)
 - Allows publisher of information to specify semantics rather than relying on consumer to assign semantics (e.g. When they read the page)
-

How?

Supplement existing markup so it can also be interpreted as RDF

- Avoid repetition
 - Built-in semantics in context with data
 - Browsers can ignore new attributes when rendering
-

Example

```
<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/"
>
<head>
<title>My home-page</title>
<meta property="dc:creator" content="Mark Birbeck"/>
<link rel="foaf:topic" href="http://www.formsPlayer.com/#us" />
</head>
<body>...</body>
</html>
```



JSON-LD

JSON-LD

JSON-LD provides a JSON syntax for RDF graphs and datasets.

JSON-LD can be used to transform JSON documents to RDF with minimal changes.

JSON-LD offers universal identifiers for JSON objects, a mechanism in which a JSON document can refer to an object described in another JSON document elsewhere on the Web.

JSON-LD

```
{
  "@context": "example-context.json",
  "@id": "http://example.org/bob#me",
  "@type": "Person",
  "birthdate": "1990-07-04",
  "knows": "http://example.org/alice#me",
  "interest": {
    "@id": "http://www.wikidata.org/entity/Q12418",
    "title": "Mona Lisa",
    "subject_of":
      "http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619",
    "creator": http://dbpedia.org/resource/Leonardo da Vinci
  }
}
```

RDF,RDFS

© Declan O'Sullivan

65

Summary of RDF Serialization

N-Triples

- RDF Test Cases, W3C Recommendation, 10 February 2004
- a **line-based, plain text serialization format** for storing and transmitting RDF data

Notation 3 (N3)

- a **shorthand non-XML serialization** of RDF models, designed with human-readability in mind
- much more compact and readable than XML RDF notation

Turtle (Terse RDF Triple Language)

- W3C Candidate Recommendation, 19 February 2013
- a **format for expressing data** in the Resource Description Framework (RDF) data model
- a subset of Notation3 (N3) language, and a superset of the minimal N-Triples format

RDF/XML

- W3C Recommendation, 10 February 2004
 - an **XML syntax** for writing down and exchanging RDF graphs
-

N-Triple

```
<http://en.wikipedia.org/wiki/Tony_Benn> <http://purl.org/dc/elements/1.1/title> "Tony Benn" .
<http://en.wikipedia.org/wiki/Tony_Benn> <http://purl.org/dc/elements/1.1/publisher> "Wikipedia" .
```

N3

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

<http://en.wikipedia.org/wiki/Tony_Benn> dc:title "Tony Benn";
dc:publisher "Wikipedia".
```

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/Tony_Benn">
    <dc:title>Tony Benn</dc:title>
    <dc:publisher>Wikipedia</dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

Turtle

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ex: <http://example.org/stuff/1.0/> .

<http://www.w3.org/TR/rdf-syntax-grammar>
  dc:title "RDF/XML Syntax Specification (Revised)" ;
  ex:editor [ ex:fullname "Dave Beckett", "Dave R. Beckett";
             ex:homePage <http://purl.org/net/dajobe/>
            ] .
```

67

Summary – basic RDF

RDF is a web standard that lets us build knowledge graphs

- Use URLs for subjects, predicates, objects with namespaces to ensure uniqueness
- Graph built from triples, but using a nested notation
- Multiple facts can be specified with minimal repetition
- Fairly straightforward for humans to write by hand

References

READ <http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/>

http://en.wikipedia.org/wiki/Resource_description_framework

<http://www.slideshare.net/lysander07/open-hpi-semweb02part2>

<http://www.slideshare.net/lysander07/open-hpi-semweb02part3>

<http://www.slideshare.net/lysander07/openhpi-24>

<http://en.wikipedia.org/wiki/Qname>

http://en.wikipedia.org/wiki/XML_Base