CHAPTER 1: Introduction and Application Scenarios

<https://euclid-project.eu/modules/chapter1.html>

Semantic web: extends the web of *documents* to a web of *data*

Apply web-based standards for encoding dataset and linking dataset to others dataset, applications can exploit data from many different sources

# Part I: Semantic Technologies and Linked Data Foundations

Google is based on text-string matching, while the technology that allows dataset to be published online and queried effectively are **semantic**: information is represented not in natural language (english, italian…) but in graph based data model, this facilitates:

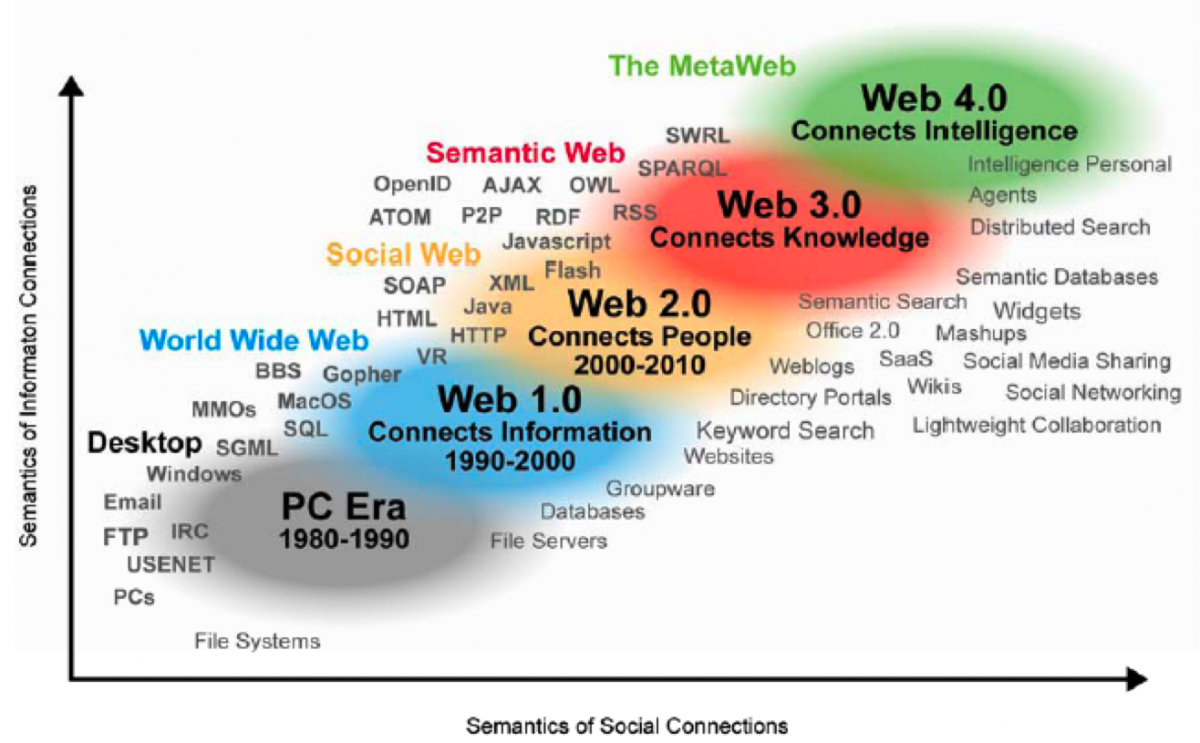
* extension
* integration
* inference
* uniform querying

text-based search are unreliable, if we are looking for a *List male british rock musicians married to scandinavias* we might get back a rock musician that plays scandinavian music

Semantic technologies resources (audio file, photograph…) can be annotated using ***Resource Description Framework* (RDF)**. In RDF formal names can be assigned to what are called *resources* (Bethoven, violino concerto, orchestra…) to types (or classes) of resources (composed, concertos…) and to relationship (*properties)* that link resources (composed-by,...)

If we apply reasoning in this way we can actually find what we are looking for.

If data from different sources are to be combined, it is important to establish links



## Web 1.0 (static)

Web èages were mostly static documents read from a server and displayed on a client, no option for users to contribute content, or for content to be tailored (fatto su misura) to a user’s specific demands

## Web 2.0 (dynamic)

Increase in use of technologies allowing the user of a browser to interact with web pages and shape their content

Made possible a wide range of social web sites

## Web 3.0 (semantic)

Computer worked only on data files in some standard logical format, not on information presented in natural language text

The initial aim of the Semantic Web was to provide standards through which people can publish documents that consist of data. Allowing programs to combine data from many dataset (just as a person can combine information from many text documents to solve a problem or perform a task)

# Ontologies

Dataset usually encode facts about individual object and events



We said in the first line that the Beatles are a music group, but we must also say the more generic fact that they are also a group, we should do this for *all* the music groups and more.

We should also add others consequences like *performing music* and *playing musical instruments*

**Ontologies**: more efficient storage and use of data by encoding *generic* facts about classes (or types of objects)



Now is sufficient to state that the Beatles (or the Clash…) are music groups and the more general fact can be derived through inference

Ontologies allows computer application to infer (dedurre) many essential facts automatically that are obvious for a person but not for a program

## HTTP

HyperText Transfer Protocol, refers to a set of conventions governing communication between a client and a server

## URI

Uniform Resource Identifier. Compact sequence of characters that identifiers an abstract or physical resource

Compact: no space character

Abstract or physical: the URI may refer to an abstract resource (like the concept *Beethoven* and *Symphony*) as well as to a document or other fire that can be retrieved from the WWW:

<http://musicbrainz.org/doc/Frequently_Asked_Questions>

scheme

authority: specifies the server where the resources is located

path: locate the resources precisely within the server’s directory structure

URIs are typically long, difficult to read and write, convenient to make use of abbreviated forms

[https://dbpedia.org/resource/Karlsruhe](https://dbpedia.org/page/Karlsruhe)

one could introduce a namespace dbp for <https://dbpedia.org/resource/> so reducing the URI to

dbp:Karlsruhe

## RDF

Resource Description Framework

Introduced originally as a data model for metadata (attributes of document, images, programs…)

statements are represented as a triple of the form

*subject--predicate--object*

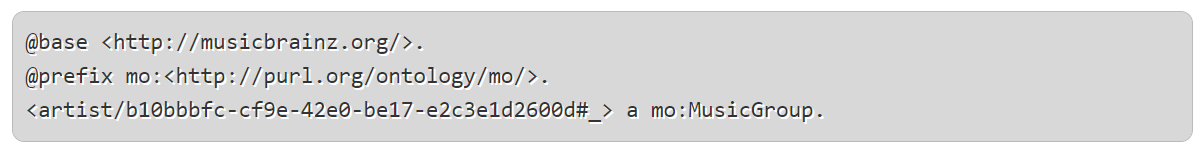
each triple is expressing a relation between the subject and the object

subject: expressed by a URI or a blank node

predicate: expressed by a URI

object: expressed by a URI or a literal (such as a number or a string)

Turtle: statements are formed by a triple followed by a full stop.

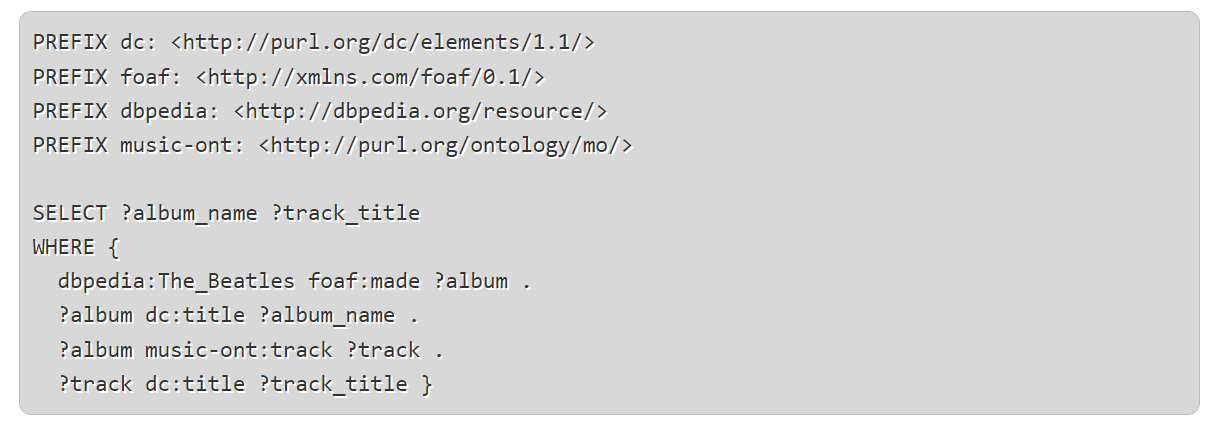


“a” is a commonly used predicate, refers to the “type” relation between a resource and its class

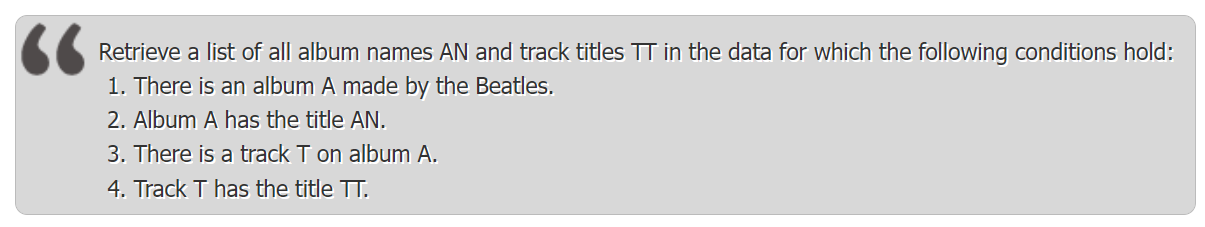
multiple statements apply to the same *subject* can be abbreviated by placing a semi-colon after the first object and then giving further predicate-object pairs separated by semicolons with a full stop after the final pair

## SPARQL

Language for formulating queries over RDF data



Translated to english we are asking:



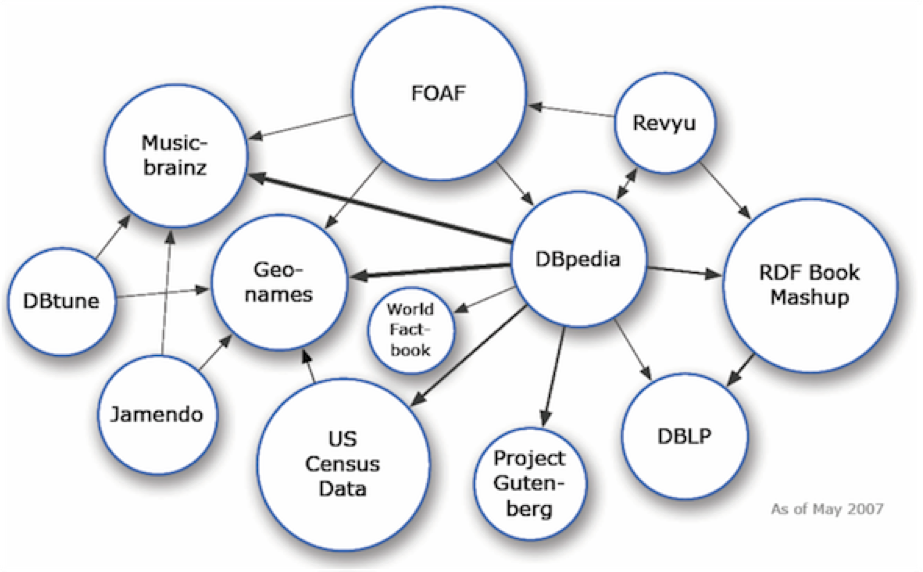
WHERE statement: list of triples

FILTER: allows conditions on the values of variables (a number should be between 1990 and 2000)

OPTIONAL: specifies data that should be retrieved *if available*

# Part II: Introduction to Linked Data

The volume of data has grown from around 2billion triples in 2007 to over 30 billion in 2011, interconnected by over 500 million RDF links



Nodes represent published dataset, links represent sets of RDF triples through which the URIs in one dataset are paired with their counterparts in another dataset

## Principles

Four simple principles for publishing data on the web, rules of best practice, not rule that must be obeyed

1. use the URIs to identify things
2. Use HTTP URIs so that people can look up those names
3. When someone looks up a URI, provide useful informations, using the standards (RDF, RDFS, SPARQL)
4. Include links to others URIs, so that they can discover more things

## Growth of linked data on the web

