

Case Study

- Post on Piazza case study folder on your team's choice for your paper/project soon.
 - You need my approval before actually working on it.
 - Please attach files in your post using menu item (Insert->Insert file) if you have any, so both your team and I can see them.
 - If multiple teams picked the same paper/project, the one posted it earliest will get it.
- One bonus point for system demonstration
 - To encourage working with real systems/tools, I'm happy to offer one bonus point for each team who can demonstrate the system associated with the paper/project during their presentation.
 - But I would recommend you only work on system demonstration after other parts of the case study are done because it might be very time consuming to get a system/tool working.
- Fill out this [spreadsheet](#) on your case study presentation time slot
 - 15 minutes for each presentation + 2-3 minutes for Q&A.

Reminder of Midterm 2 Next Week

- Midterm covers chapters 8-13. 30 questions and 0.5 for each question. Total mark: 15
 - We will still have multi-choice questions (pick one from multiple circles), multi-answer questions (pick two or more from multiple squares) and short answer questions
- The exam will be available 4:30-5:45 PM ET Thursday, November 19th on Blackboard
 - No class after exam
- Send me on Piazza via **private posts** on at most five recommended questions by the end of Wednesday, November 18th
- Using the lockdown browser, blackboard will record your picture, your photo ID and your activity during exam
- If you have difficulty to access the exam at blackboard, you can contact me via slack or email
- During exam, it is difficult to discuss questions with me. Just answer the question based on your understanding. If you have questions, you can contact me after exam
- Because we are having midterm next week, the deadline for this chapter's exercise is Wednesday after midterm 2
 - Working on exercise should still help your exam preparation

IS 651: Distributed Systems

Chapter 12: Semantic Web

Jianwu Wang

Fall 2020

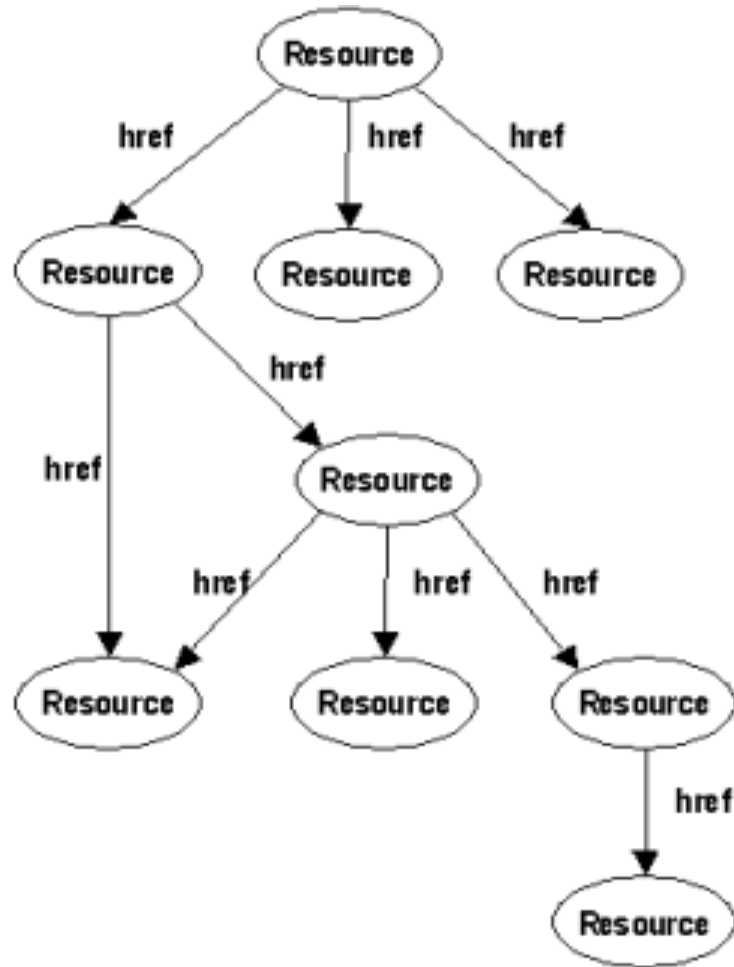
Learning Outcomes

- After learning this chapter, you should be able to
 - Understand the basics of Semantic Web and its techniques
 - Be able to do simple SPARQL SELECT queries

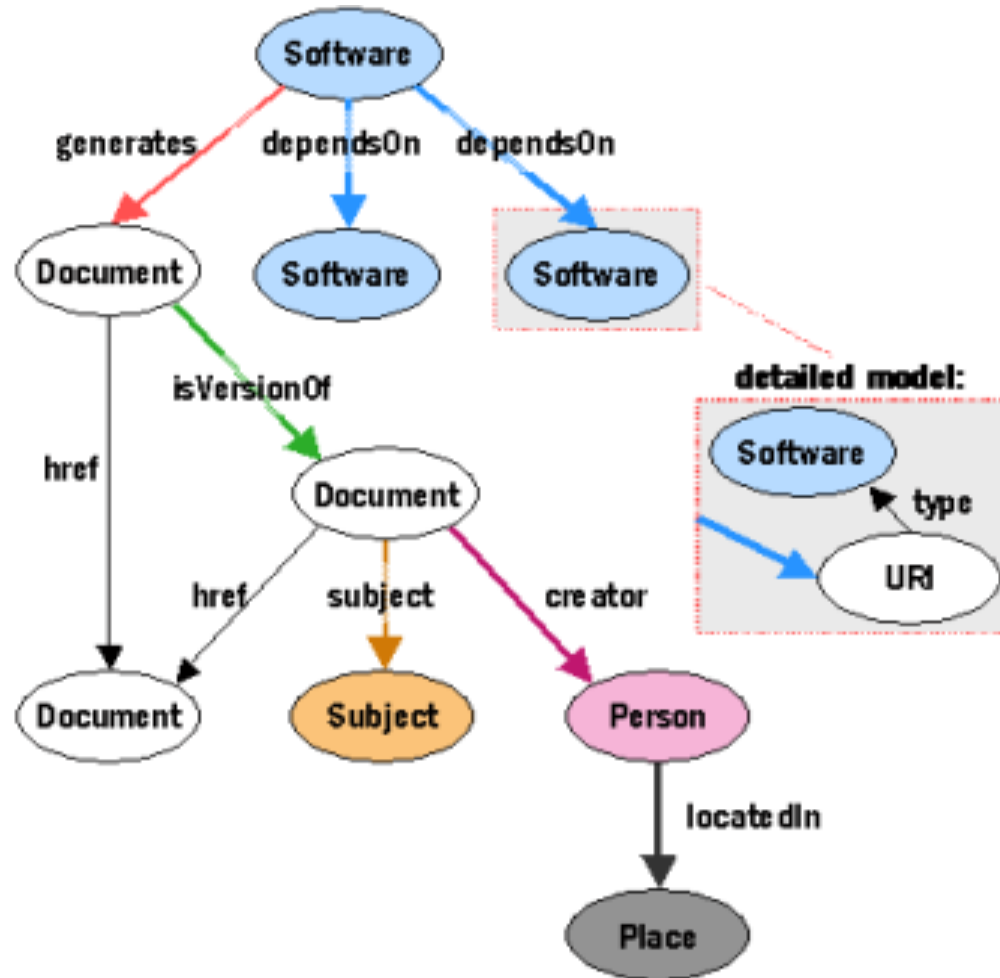
The Semantic Web

- Syntax VS Semantics
- Web 3.0: Make the web easier to be understood by computers
 - Web 2.0: user generated web contents, usability, and interoperability. Examples are blog, wiki, social network sites and mashup.
- Techniques/standards for Semantic Web
 - Resource description framework (RDF) - describing data with a global naming scheme (URIs)
 - RDF Schema (RDFS) - A standard way of describing the properties of that data
 - Web Ontology Language (OWL) - A standard way of describing relationships between data items.
 - RDF Stores and SPARQL - A standard way of storing and querying graphs.

From Current Web to Semantic Web



a) Current Web



b) Semantic Web

Resource Description Framework (RDF)

- [RDF](#) is a graph model built by triples: <resource/subject, property/predicate, value/object>
- RDF describes data with a global naming scheme (URIs)
- RDF is independent of any serialization
- Common RDF serialization notation/language are [N3](#) and XML
- Commonly used RDF vocabularies include [Dublin Core](#), [FOAF](#)
 - You can define your own vocabularies
- [W3C's RDF Validation Service](#) for validation and graph representation
 - [One graph representation example](#)

N3 and XML Serialization for RDF

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema> .  
  
<http://userpages.umbc.edu/~jianwu/foaf.rdf>  
foaf:name  
"Jianwu Wang"^^xsd:string .
```

```
<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://userpages.umbc.edu/~jianwu/foaf.rdf">  
    <foaf:name>Jianwu Wang</foaf:name>  
  </rdf:Description>  
</rdf:RDF>
```


RDFSchema

- RDF Schema is an extension of the basic RDF vocabulary
- It provides a standard way of describing the properties of that data
- It adds the concept of class (sub-class), and associated properties (sub-property) to RDF triples

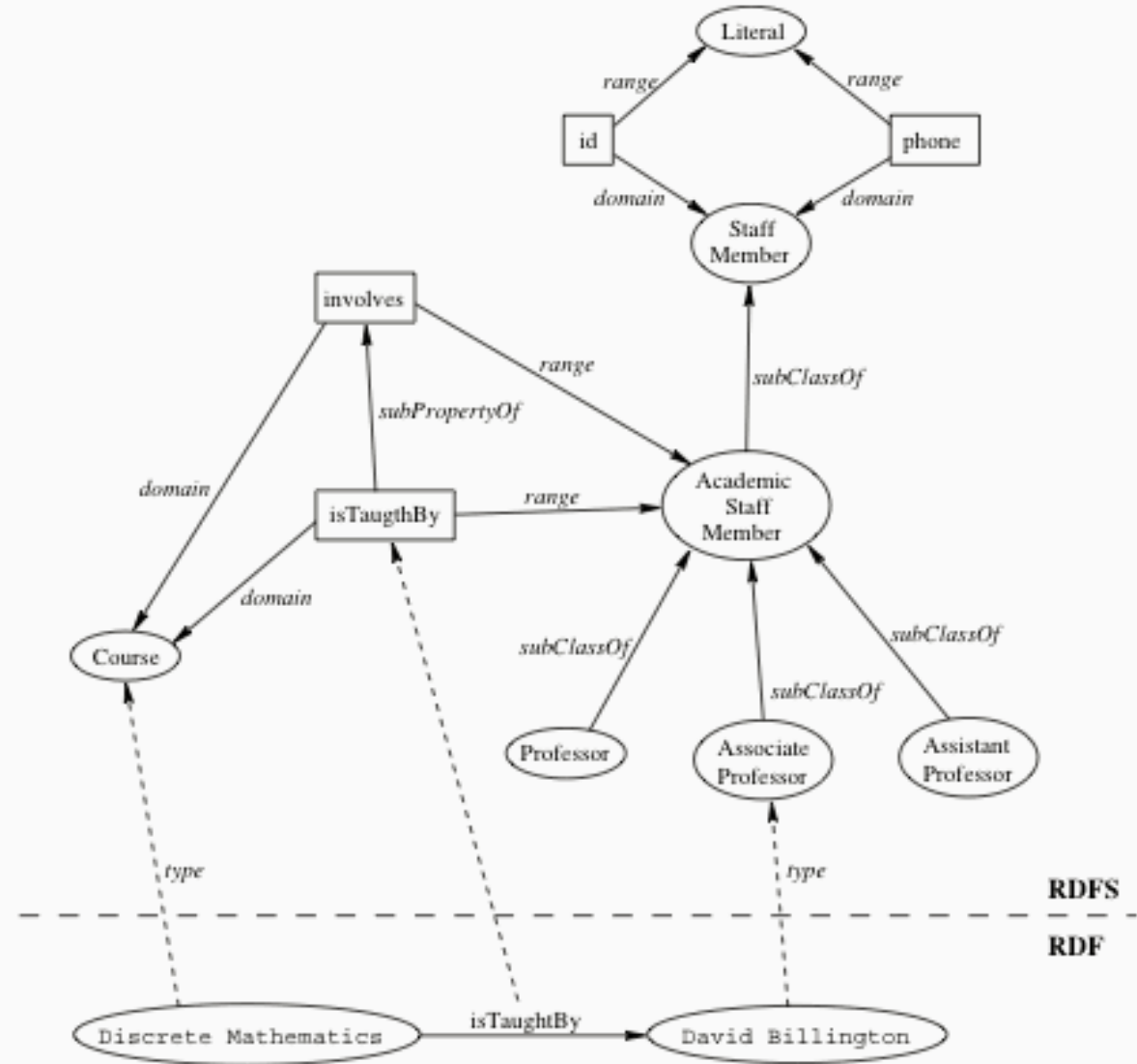


Figure 3.6 RDF and RDFS layers

RDFSchema (2)

- The RDF Schema class and property system is similar to the type systems of object-oriented programming languages such as the Java language.
- A difference is RDF Schema defines a property separately/independently by its range class and domain class
- A specific resource is a type of a class
 - UMBC is a type of university class
- You can find RDFSchema example in N3 at book chapter

Quick Question

- Extend the RDFShema example to add student concept and instances

OWL (Web Ontology Language)

- [OWL](#) is a language for defining and instantiating Web ontologies
- Ontology is a term borrowed from philosophy that refers to the science of describing the kinds of entities in the world and how they are related
- An OWL ontology may include descriptions of classes, properties and their instances
- Given such an ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but **entailed** by the semantics

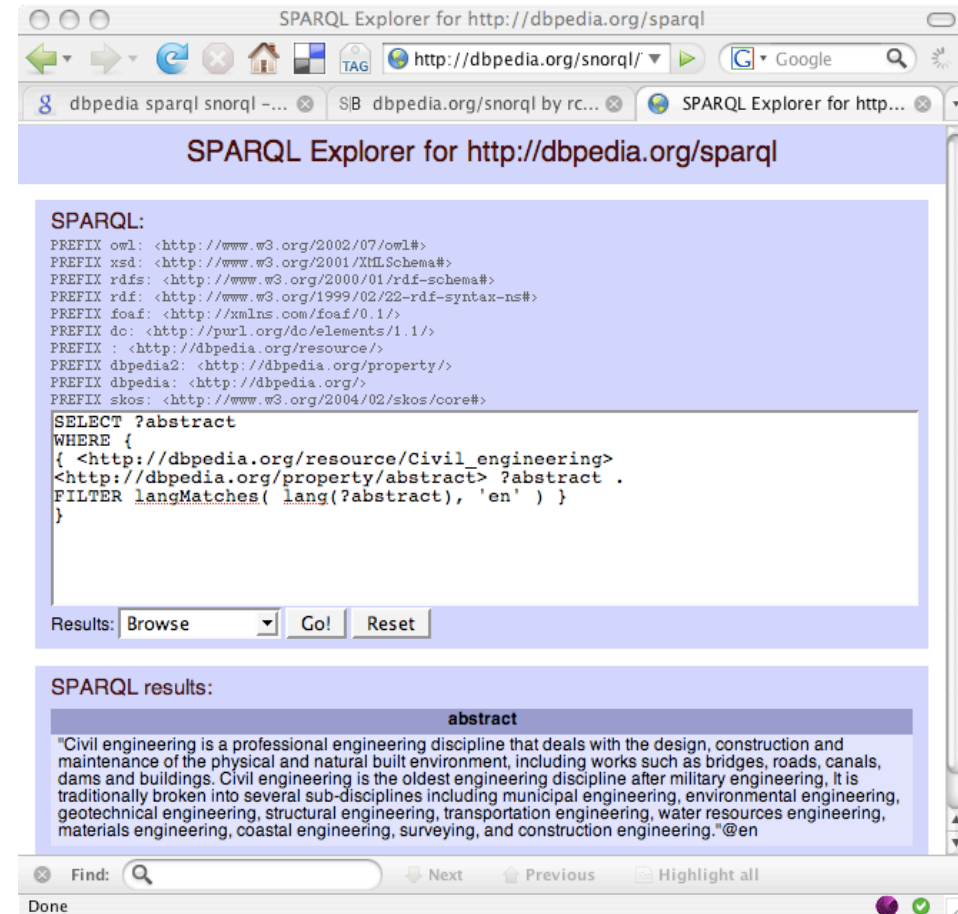
SPARQL

- [SPARQL](#) is a query language for RDF
 - Recursive acronym: SPARQL Protocol and RDF Query Language
- Four types of SPARQL queries
 - *SELECT*
 - *CONSTRUCT*
 - *ASK*
 - *DESCRIBE*
- You can specify one or more SPARQL endpoints in SPARQL query
 - Endpoint list: <https://www.w3.org/wiki/SparqlEndpoints>

SPARQL Demo

- [SPARQL Explorer](#)
- An SELECT query example

```
PREFIX dbo: <http://dbpedia.org/property/>
SELECT ?name ?birth
{
?person dbo:birthPlace :Berlin .
?person dbo:birthDate ?birth .
?person foaf:name ?name .
?person dbo:deathDate ?death .
FILTER (?birth < "1900-01-01"^^xsd:date) . } ORDER
BY ?name
#People who were born in Berlin before 1900
```



The screenshot shows a web browser window titled "SPARQL Explorer for http://dbpedia.org/sparql". The address bar shows "http://dbpedia.org/snorql/". The main content area displays the following SPARQL query:

```
SPARQL:
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX : <http://dbpedia.org/resource/>
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbpedia: <http://dbpedia.org/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT ?abstract
WHERE {
{ <http://dbpedia.org/resource/Civil_engineering>
<http://dbpedia.org/property/abstract> ?abstract .
FILTER langMatches( lang(?abstract), 'en' ) }
}
```

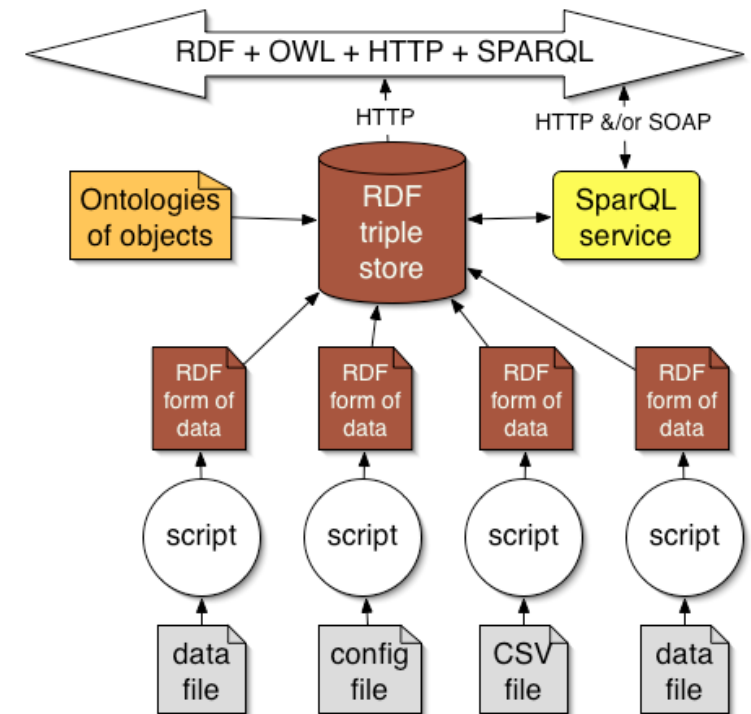
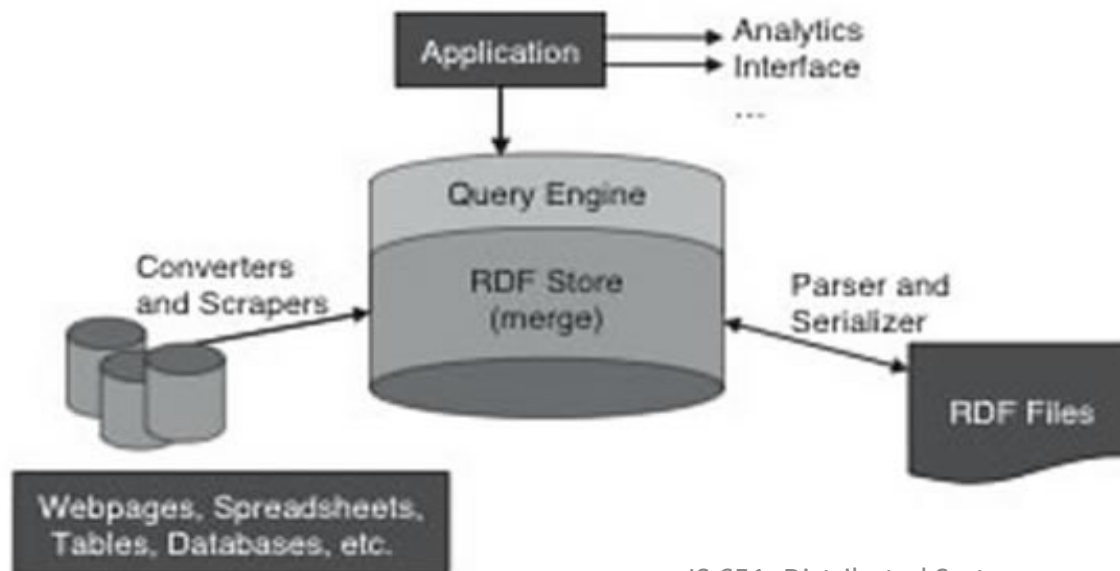
Below the query, there are buttons for "Results: Browse", "Go!", and "Reset". The "SPARQL results:" section shows a table with one row:

abstract
"Civil engineering is a professional engineering discipline that deals with the design, construction and maintenance of the physical and natural built environment, including works such as bridges, roads, canals, dams and buildings. Civil engineering is the oldest engineering discipline after military engineering, it is traditionally broken into several sub-disciplines including municipal engineering, environmental engineering, geotechnical engineering, structural engineering, transportation engineering, water resources engineering, materials engineering, coastal engineering, surveying, and construction engineering."@en

At the bottom of the browser window, there is a search bar with "Find:" and a magnifying glass icon, and buttons for "Next", "Previous", and "Highlight all". The status bar at the very bottom shows "Done" and a green checkmark.

RDF (Triple) Stores

- An RDF triple store is a purpose-built **graph** database for the storage and retrieval of RDF metadata
- RDF Store Software: [Jena](#), [RDF4J/Sesame](#), [Neo4j](#), etc.



create RDF files from other formats

Linked Data

- A method of publishing and interlinking structured data so it can become more useful through semantic queries
- It builds upon HTTP, RDF and URIs
- This enables data from different sources to be connected and queried
- Famous linked data sets
 - [DBpedia](#) for extracted data from [Wikipedia](#)
 - [FOAF](#) for persons, their properties and relationships

Semantic Web Criticism

- Web is too vast and vague
 - But what about specific areas?
- Too computer-friendly



"Now! ... *That* should clear up
a few things around here!"