

Semantic Web, Linked Data, RDF e JSON-LD

Introduzione

- **Semantic Web (SW):** Estensione del World Wide Web che mira a rendere le informazioni comprensibili anche dalle macchine, separando i dati dalla loro rappresentazione.
- **Linked Data:** Dati strutturati interconnessi con altri dati, estensione delle tecnologie Web standard (HTTP, RDF, URI) per condividere informazioni interpretabili da agenti automatici.
- **Linked Open Data (LOD):** Linked Data accessibili e interrogabili pubblicamente.
- **Resource Description Framework (RDF):** Metodo per la descrizione concettuale o la modellazione di informazioni, basato su triple soggetto-predicato-oggetto.
- **JSON-LD:** Metodo per codificare i Linked Data usando JSON.

Semantic Web Stack

Architettura del Web Semantico, organizzata in livelli:

1. **Risorse e URI:** Identificazione delle risorse tramite URI e descrizione in linguaggio naturale.
2. **Linguaggi di Markup:** Creazione di documenti con dati strutturati (es. XML).
3. **Standard per lo scambio di informazioni:** Rappresentazione delle informazioni come grafo (es. RDF).
4. **Standard per tassonomie:** Rappresentazione di proprietà e categorie (es. RDFS).
5. **Ontologie e Regole:** Definizione di concetti, ontologie (es. OWL) e regole (es. RIF/SWRL) per la semantica dei dati, e strumenti per l'estrazione di informazioni (es. SPARQL).
6. **Logica e Inferenza:** Programmi e umani eseguono operazioni logiche per la verifica dei fatti e l'inferenza.

Tecnologie Chiave

- **RDF:** Modello per definire link etichettati tra risorse (dati RDF).
- **OWL:** Linguaggio per creare ontologie, con vocabolario di concetti e relazioni, utilizzabile anche per validare dati RDF.
- **SPARQL:** Linguaggio per interrogare dataset di dati RDF (triplestore).

Linked Data

- Dati strutturati interconnessi con altri dati.
- Estensione di tecnologie Web standard (HTTP, RDF, URI).
- Condivisione di informazioni interpretabili da agenti automatici.
- Database RDF del Semantic Web.

Resource Description Framework (RDF)

- **Triple:** Affermazioni sulle risorse nella forma soggetto-predicato-oggetto.
 - **Soggetto:** Risorsa (URI).
 - **Oggetto:** Risorsa (URI) o letterale (stringa, numero, data, ecc.).
 - **Predicato:** Relazione tra risorse, una proprietà (URI).
- **Esempio:** "Umberto Eco è autore de Il Nome della Rosa"
 - Soggetti: Umberto Eco (URI), Il Nome della Rosa (URI).
 - Predicati: è un, si chiama, ha scritto (URI).

- Oggetti: autore, "Umberto Eco", Il Nome della Rosa (URI), libro, "Il Nome della Rosa".

Diventa, usando i URI corretti:

- `<https://dbpedia.org/umberto_eco> rdf:type foaf:person .`
- `<https://dbpedia.org/umberto_eco> foaf:name "Umberto Eco".`
- `<https://dbpedia.org/umberto_eco> foaf:made`
`<http://www.anobii.com/books/Il_nome_della_rosa> .`
- `<http://www.anobii.com/books/Il_nome_della_rosa> rdf:type bibo:Book .`
- `<http://www.anobii.com/books/Il_nome_della_rosa> dc:title "Il Nome della Rosa".`

- **Risorse anonime (Blank node):** Risorse variabili (esistenziali), non costanti, senza un URI specifico.
- **Grafi RDF:** Insieme di triple RDF, rappresentabili come reti semantiche.
 - Nodi: Risorse (soggetti e oggetti).
 - Archi: Predicati.
- **Ontologie:** tutti i predicati, i soggetti e gli oggetti (che non siano stringhe) sono definite in ontologie, insieme di triple RDF che definiscono il significato di ogni termine.
- **Vocabolario Base RDF:** vocabolario di base definito dall'ontologia RDF.

Vantaggi e Svantaggi di RDF

- **Vantaggi:**
 - Modello a triple semplice e minimalista.
 - Struttura dati a grafo.
 - Modularità: elaborazione parallelizzabile, output coerente anche con informazioni parziali.
 - Combinato con OWL, permette il ragionamento e l'estrazione efficiente di informazioni (grazie alla Description Logic).
- **Svantaggi:**
 - Dati frammentati in piccole unità.
 - Difficoltà nella descrizione di relazioni N-arie.
 - Difficoltà nell'attribuire informazioni alle triple (reificazione, Named Graphs come soluzioni parziali).

Relazioni N-arie e Reificazione

- **Relazioni N-arie:** Esempio: "Umberto Eco ha scritto 'Il nome della rosa' nel 1980".
 - Soluzione: Creazione di un evento di creazione (blank node) che collega l'autore, il libro e l'anno.
- **Reificazione:** Esempio: "Wikipedia dice che Umberto Eco ha scritto 'Il nome della rosa'".
 - Rappresentazione della tripla come entità autonoma.
 - Utilizzo di classi e predicati RDF predefiniti (rdf:Statement, rdf:subject, rdf:predicate, rdf:object).
 - La tripla di base non è asserita (non si trova cercando "Chi ha scritto il nome della rosa?").
- **Named Graphs:**
 - Assegnazione di un nome a un grafo di triple.
 - La tripla interna è in un grafo separato con identità.
 - La frase esterna punta al grafo separato.

- La tripla di base può essere presente o meno (dipende dal motore).

Confronto tra Modelli di Dati

- **Tabellare (es. SQL):**
 - Efficiente per dati regolari e prevedibili (schema).
 - Tabelle di collegamento (join) per relazioni.
 - Difficoltà con schemi irregolari o dati sparsi.
- **Gerarchico (es. JSON):**
 - Nessun bisogno di schemi.
 - Gestione di difformità tra entità.
 - Efficiente per relazioni 1-N.
 - Difficoltà con relazioni M-N (simulazione manuale).
- **Grafo (es. RDF):**
 - Nessun bisogno di schemi.
 - Gestione di difformità tra entità.
 - Efficiente per relazioni 1-N e M-N (scomposizione in relazioni 1-a-molti).
 - Adatto per dati irregolari e inferenze.

Serializzazioni RDF

Formati per la rappresentazione di dati RDF:

- **RDF/XML:** Sintassi basata su XML, primo formato standard.
- **Turtle e Trig:** Formati compatti e human-friendly.
- **JSON-LD:** Serializzazione basata su JSON.
- **RDFa e Microdata:** Embedding di triple RDF in HTML.

Esempi di Serializzazione

Esempio 1: "Umberto Eco è autore de Il Nome della Rosa"

- **Turtle:**

```
@prefix anobii: <http://www.anobii.com/books/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix db: <https://dbpedia.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

db:umbertoeco rdf:type foaf:person ;
    foaf:name "Umberto Eco" ;
    foaf:made anobii:Il_nome_della_rosa .

anobii:Il_nome_della_rosa a <http://purl.or/ontology/bibo/Book> ;
    dc:title "Il nome della rosa" .
```

- **RDF/XML:**

```
<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/terms/">
<foaf:person rdf:about="https://dbpedia.org/umbertoeco">
  <foaf:name>Umberto Eco</foaf:name>
  <foaf:made>
    <rdf:Description
rdf:about="http://www.anobii.com/books/Il_nome_della_rosa">
      <dc:title>Il nome della rosa</dc:title>
      <rdf:type rdf:resource="http://purl.org/ontology/bibo/Book"/>
    </rdf:Description>
  </foaf:made>
</foaf:person>
</rdf:RDF>

```

- **JSON-LD:**

```

[
  {
    "@id": "https://dbpedia.org/umbertoeco",
    "@type": "http://xmlns.com/foaf/0.1/person",
    "http://xmlns.com/foaf/0.1/name": "Umberto Eco",
    "http://xmlns.com/foaf/0.1/made": {
      "@id": "http://www.anobii.com/books/Il_nome_della_rosa",
      "@type": "http://purl.org/ontology/bibo/Book",
      "http://purl.org/dc/terms/title": "Il nome della rosa"
    }
  }
]

```

- **RDFa:**

```

<html xmlns:anobii="http://www.anobii.com/books/"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:dc="http://purl.org/dc/terms/"
xmlns:db="https://dbpedia.org/"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<p about="db:umbertoeco" typeof="foaf:person">
  <span property="foaf:name">Umberto Eco</span>
  <span rel="foaf:made">è autore de
    <span resource="anobii:Il_nome_della_rosa">
      <span property="dc:title">Il Nome della Rosa</span>
    </span>
  </span>
</p>

```

- **Microdata:**

```

<p itemscope
  itemid="https://dbpedia.org/umbertoeco"
  itemtype="http://xmlns.com/foaf/0.1/person">
  <span itemprop="name">Umberto Eco</span>

```

```
<span itemprop="made">è autore de  
  <span itemscope  
    itemid="http://www.anobii.com/books/Il_nome_della_rosa">  
      <span itemprop="http://purl.org/dc/terms/title">Il Nome della Rosa</span>  
    </span>  
  </span>  
</p>
```

(Gli altri esempi seguono logiche simili, con sintassi specifiche per ciascun formato)

RDFS e OWL

- **RDF Schema (RDFS):** Insieme di classi e regole di ragionamento per generare nuova conoscenza (inferenze) da dati RDF.
- **Web Ontology Language (OWL):** Estende RDFS con regole di inferenza più ricche e regole di coerenza per segnalare contraddizioni.

SPARQL

- **SPARQL Protocol and RDF Query Language:** Linguaggio per interrogare dataset RDF e restituire tabelle di dati.
 - usa le triple RDF per cercare dati nel database.
 - i risultati sono insiemi di triple, o liste di valori.
- **Esempio:**

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
PREFIX bibo: <http://purl.org/ontology/bibo/>  
PREFIX dc: <http://purl.org/dc/terms/>
```

```
SELECT ?name ?title  
WHERE {  
  ?p a foaf:Person .  
  ?p foaf:name ?name .  
  ?p foaf:made ?b .  
  ?b a bibo:Book .  
  ?b dc:title ?title .  
}
```

- **SPARQL Update:** Sintassi simile a SPARQL per la manipolazione di dati RDF (inserimenti, modifiche, cancellazioni).

Conclusioni

- Il Semantic Web è una disciplina complessa ma potente.
- Permette di rappresentare grandi quantità di dati in modo destrutturato e riorganizzabile.
- La struttura a grafo e la possibilità di inferenze rendono il SW significativo per applicazioni sofisticate.