



Rappresentare,
pubblicare e usare i
Linked Data

Roadmap

1. Requirements: analizzare la funzione e l'uso dell'ontologia (vedere metodologia NEON)
2. Modellazione: creare l'ontologia e codificarla in OWL
3. Knowledge graph: descrivere i dati con il vocabolario creato
4. Pubblicazione dei dati: caricamento su una LDP
5. Applicazione: Progettare (e sviluppare) l'applicazione lato client

1. Requirements

- Raccogliere e analizzare la documentazione sul dominio di conoscenza
 - Siti
 - Manuali
 - Standard
- Individuare
 - Funzionalità: reference (es. Cidoc-CRM), annotazione dati (es. FoodOn ontology), interoperabilità (MobiVoc: Open Mobility Vocabulary), accesso ai dati (es. Music Brainz).
 - Tipo di utenti: specialisti, pubblico generico, comunità specifiche, ecc.
- *Competency questions* per esprimere le domande a cui l'ontologia deve rispondere
 - spesso espresse in SPARQL (e poi riviste) dall'inizio
 - *Con quali artisti ha collaborato un certo gruppo?*
 - *Quali innovazioni ha prodotto una certa tecnologia?*
 - *Quale combinazione di ingredienti è adatta per una determinata intolleranza?*
 - *Dove si getta un involucro di un certo tipo?*
 - ...

2. Modellazione



Ontology
engineering

Utilizzo di pattern
(eventi, liste, ecc.)
si possono duplicare o
importare



Altre ontologie /
sorgenti di
conoscenza

Domain level
Fondazionali
Vocabolari e
tassonomie



Allineamento

OWL
RDF
SKOS
Importazione di
ontologie



Ragionamento automatico



Documentazione

Documentazione

- Visualizzazione
 - Plugin Protégé
 - Software web-based
- Documentazione
 - LOD (<http://www.essepuntato.it/lode>)
 - rdfs:label e rdfs:comment + language tag
- Metadati obbligatori dell'ontologia
 - <dc:description> Descrizione dell'ontologia </dc:description>
 - <rdfs:label> Label ontologia </rdfs:label>
 - <vann:preferredNamespacePrefix>...</...>
 - <vann:preferredNamespaceUri rdf:resource="http://...">
 - <dc:date>2018-11-15</dc:date>
 - <dc:creator> ... </dc:creator>
 - <dc:title>...</dc:title>

2. Modellazione: altre ontologie e vocabolari

- Commonsense / Top level
 - <https://schema.org>
 - Dbpedia (<http://dbpedia.org/ontology/>) e Wikidata
 - Basic Formal Ontology (<http://basic-formal-ontology.org>)
 - Dolce (DUL: http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS_Ultralite)
- Mid-level
 - Prov, Dublin Core, Foaf, Geo, Time, Bibo, Lode, Event, OA, ecc.
- Repository di vocabolari:
 - EU Core Vocabularies on Joinup <https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/core-vocabularies>
 - Linked Open Vocabularies <https://lov.linkeddata.es/dataset/lov/>
 - <https://www.bbc.co.uk/ontologies>

3. Knowledge graph

Design degli IRI:

- Convenzioni per classi, proprietà e individui
- Creare *fragment identifier* comprensibili (oltre alle label)

Popolamento dell'ontologia

- Inserimento manuale
- Importazione da mapping tools (da csv, tsv, excel, ecc.) es. Open Refine (Google)

Visualizzazione di esempi significativi

- vedere indicazioni per il progetto

3.1 Importazione dei dati



- Importazione tramite SPARUL
 - Per via programmatica
 - Importazione diretta via query federate da repositories pubbliche (commonsense/enciclopedica o domain specific)
 - Importazione da Google Knowledge Graph
- Importazione da un data base via R2RML

4. Pubblicazione



- LDP locale
 - Virtuoso
 - GraphDB
 - Blazegraph
 - Architettura Jena + Pellet/ Jena+ Fuseki (endpoint SPARQL)
- Materializzazione delle inferenze
- Upload
 - Creazione e configurazione del repository
 - Caricamento dei dati
 - Configurazione prefissi

5. Design applicazioni

- Definire funzionalità e flusso
 - Flow chart interazione con l'utente
 - Mockup interfacce
- Progettare e testare le query
 - Integrazione con LOD (query su sorgenti di dati diverse)
- Implementazione dell'applicazione (client)