

Relazione Agrievents

Obbiettivo

Realizzare una piattaforma web che sfrutti le tecnologie offerte dal web semantico, che si occupi di modellare con ontologie appropriate eventi agricoli e offerte prodotte da agricoltori. Inoltre devono essere memorizzate le informazioni relative alla data, posizione e luogo dell'evento. In un offerta sarà presente frutta di vario tipo mentre agli organizzatori sarà permesso inserire eventi mercato con i relativi dati connessi. Gli agricoltori potranno inserire offerte scegliendo un particolare evento mercato e infine sarà possibile visualizzare eventi mercato che mettono a disposizione sotto forma di offerta un certo tipo di frutta.

Strumenti utilizzati

E' stato adoperato **Protégé 5.0.0 beta-17** per realizzare una bozza del modello. Inoltre si è reso necessario individuare diverse ontologie utili al fine di modellare in modo conciso e ordinato il concetto di evento mercato agricolo. Sono state utilizzate le ontologie:

- **The GoodRelations Ontology** (<http://www.heppnetz.de/ontologies/goodrelations/v1.html>)
- **DBpedia Ontology** (<http://dbpedia.org/resource/>)
- **Programme Location Core Vocabulary** (<http://www.w3.org/ns/locn>)
- **The Event Ontology** (<http://purl.org/NET/c4dm/event.owl>)
- **The Time Ontology** (<http://www.w3.org/2006/time>)
- **The Basic GeoOntology** (http://www.w3.org/2003/01/geo/wgs84_pos)
- **The FOAF Ontology** (<http://xmlns.com/foaf/0.1/>)

Come server/database per caricare le asserzioni modellate con Protegè è stato usato Dydra, qui è stato creato un account per accedere ai servizi offerti dallo stesso Dydra.

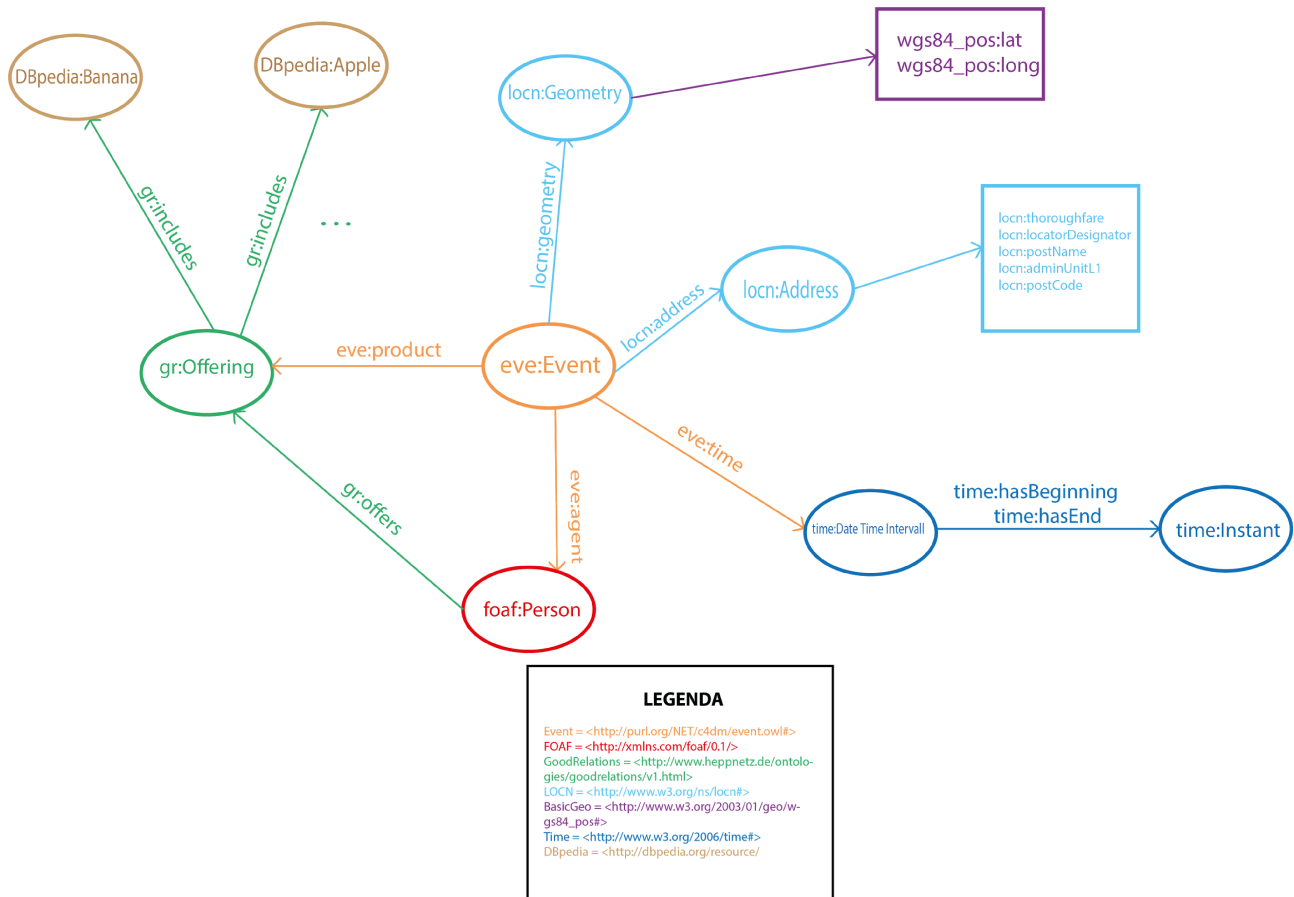
Javascript è stato il linguaggio utilizzato per generare dinamicamente le pagine interrogando il triplestore risiedente in Dydra.

Modellazione

Per modellare il concetto di evento si è fatto uso della classe Event, ogni Event è individuato da un Address che contiene le informazioni circa il suo indirizzo. La classe Geometry indica la sua posizione geospaziale quindi la posizione eventualmente visualizzabile in una mappa con l'ausilio di webservice esterni. In Address sono contenute tutte le informazioni inerenti all'indirizzo, CAP, numero civico, ecc...

Tramite la classe **Person** presa da FOAF è stato modellato il concetto di Agricoltore. Un Agricoltore può vendere frutta tramite il predicato **offers** e mettere così un'offerta (di classe **Offering**) a disposizione in un particolare mercato tramite il predicato **product**. L'offerta inserita nel triplestore tramite il predicato **includes** può includere diversi tipi di frutta reperiti nel nostro caso da DBpedia (**Banana**, **Apple**, ecc ...).

Infine un'evento deve essere ben localizzato anche temporalmente con una data di inizio evento e una di fine evento, per fare ciò si è fatto uso del vocabolario offerto dall'ontologia Time che mette a disposizione degli oggetti di classe **DateTimeInterval** che sono legati all'evento tramite il predicato **time**, ogni oggetto di questo tipo tramite i predicati **hasBeginning** e **hasEnd** è legato ad un **Instant** che contiene data e ora di inizio e fine di un certo evento. L'istante di tempo ha il tipo di una stringa ed inoltre ha un determinato formato. Se non viene rispettato tale formato la chiamata a metodi che ne fanno uso (ad esempio per filtrare i dati selezionati tramite una query sparql) potrebbe dare risultati inattesi.



Esempio di modellazione

Abbiamo modellato il seguente esempio:

- L'agricoltore **Mario_Rossi** offre una **Vendita_Mario_Rossi_Apple** al **Mercato_Catania**
- Il **Mercato_Catania** ha un **Giorno_1_Mercato_Catania**
- La **Vendita_Mario_Rossi_Apple** include **Apple**
- **Giorno_1_Mercato_Catania** ha un **Inizio_Mercato_Catania**
- **Giorno_1_Mercato_Catania** ha un **Fine_Mercato_Catania**
- **Mercato_Catania** ha agente **Mario_Rossi**
- **Mercato_Catania** produce **Vendita_Mario_Rossi_Apple**

Classi

Le classi principali utilizzate sono:

- **foaf:Person:** Mario_Rossi
- **gr:Offering:** Vendita_Mele
- **locn:Address:** Indirizzo_Mercato_Catania
- **locn:Geometry:** Mercato_Catania_Geometry
- **eve:Event:** Mercato_Catania

Proprietà

Le proprietà utilizzate sono:

Proprietà	Domain	Range
gr:offers	foaf:Person	gr:Offering
eve:product	eve:Event	gr:Offering
eve:agent	eve:Event	foaf:Person
gr:includes	gr:Offering	"DBpedia"
locn:geometry	eve:Event	locn:Geometry
locn:address	eve:Event	locn:Address
time:time	eve:Event	time:DateTimeInterval
time:hasBeginning	time:DateTimeInterval	time:Instant
time:hasEnd	time:DateTimeInterval	time:Instant

Di seguito alcuni frammenti del file xml che modella la nostra ontologia:

```
<owl:NamedIndividual rdf:about="&agreve;Mercato_Catania">
  <rdf:type rdf:resource="&eve;Event"/>
  <locn:address>
    <locn:Address rdf:about="&agreve;Mercato_Catania_Address">
      <rdfs:label>Piazza Stesicoro, 37, 95131, Catania, Italia</rdfs:label>
      <locn:fullAddress>Piazza Stesicoro, 37, 95131, Catania,
Italy</locn:fullAddress>
      <locn:thoroughfare>Piazza Stesicoro</locn:thoroughfare>
      <locn:locatorDesignator>37</locn:locatorDesignator>
      <locn:postCode>95131</locn:postCode>
      <locn:postName>Catania</locn:postName>
      <locn:adminUnitL1>IT</locn:adminUnitL1>
    </locn:Address>
  </locn:address>
  <locn:geometry>
    <locn:Geometry rdf:about="&agreve;Mercato_Catania_Geometry">
      <wgs84_pos:lat>37.507713</wgs84_pos:lat>
      <wgs84_pos:long>15.086394</wgs84_pos:long>
    </locn:Geometry>
  </locn:geometry>
```

```

    <rdfs:label xml:lang="en">Catania Market</rdfs:label>
    <rdfs:label xml:lang="it">Mercato Catania</rdfs:label>
    <rdfs:comment xml:lang="en">A place where you can buy whatever you
want.</rdfs:comment>
    <rdfs:comment xml:lang="it">Un posto dove puoi comprare qualsiasi
cosa vuoi.</rdfs:comment>
    <eve:time rdf:resource="&agreve;Giorno_1_Mercato_Catania"/>
        <eve:agent rdf:resource="&agreve;Antonio_Bianchi"/>
        <eve:agent rdf:resource="&agreve;Elena_Neri"/>
        <eve:agent rdf:resource="&agreve;Giuseppe_Verdi"/>
        <eve:agent rdf:resource="&agreve;Mario_Rossi"/>
    <eve:product
rdf:resource="&agreve;Vendita_Antonio_Bianchi_Common_fig"/>
        <eve:product rdf:resource="&agreve;Vendita_Elena_Neri_Apple"/>
        <eve:product rdf:resource="&agreve;Vendita_Mario_Rossi_Apple"/>
        <eve:product rdf:resource="&agreve;Vendita_Giuseppe_Verdi_Pear"/>
    </owl:NamedIndividual>
.
.
.

<owl:NamedIndividual rdf:about="&agreve;Mario_Rossi">
    <rdf:type rdf:resource="&foaf;Person"/>
    <gr:offers rdf:resource="&agreve;Vendita_Mario_Rossi_Strawberry"/>

```

```

    <gr:offers rdf:resource="&agreve;Vendita_Mario_Rossi_Apple"/>
    <gr:offers rdf:resource="&agreve;Vendita_Mario_Rossi_Grape"/>
</owl:NamedIndividual>

```

Libreria sparql

E' stato utilizzata la sparql suite (http://www.dmi.unict.it/~longo/sparql_suite/). Tale libreria mette a disposizione una funzione `sparql_query` che interroga il server tramite una query passata come parametro. La funzione `sparql_query` richiede un oggetto "processor(table, summary)" definito nella sparql suite, e un end point dove effettuare l'interrogazione con la query stabilita. Qui di seguito un esempio di un suo utilizzo:

```

sparql_query("http://dydra.com/info-agrievents/agrievents/sparql", new
Processor(tableResults2, summary));

```

Query

Qui di seguito sono mostrate le query più rappresentative del sistema modellato:

Permette la selezione degli eventi mercato, degli agricoltori presenti e della frutta che vende ogni agricoltori.

```

PREFIX eve:<http://purl.org/NET/c4dm/event.owl#>
PREFIX gr:<http://purl.org/goodrelations/v1#>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
PREFIX time:<http://www.w3.org/2006/time#>
PREFIX dc:<http://purl.org/dc/elements/1.1/>
PREFIX locn:<http://www.w3.org/ns/locn#>
SELECT DISTINCT ?event ?di ?address ?person ?offering ?fruit where {
    ?event a eve:Event .
    ?offering a gr:Offering .
    ?person a foaf:Person .

```

```

?fruit a [] .
?addr a locn:Address .
?event eve:product ?offering .
?person gr:offers ?offering .
?offering gr:includes ?fruit .
?event locn:address ?addr .
?addr locn:fullAddress ?address .
?event eve:time ?time .
?time time:hasBeginning ?datei .
?datei dc:date ?di
} ORDER BY ?di ?event ?person

```

Permette la selezione della frutta che contiene una determinata sottostringa.

```

select distinct ?uri{
  ?uri a ?c .
  FILTER (regex(STR(?uri), "http://dbpedia.org/resource/")) .
  FILTER (regex(STR(?uri), "ap", "i"))
}

```

Con una combinazione delle due precedenti query, si permette all'utente di cercare della frutta e di conseguenza trovare un evento mercato che includa un'offerta contenente la frutta desiderata.

La query seguente invece permette l'inserimento di un evento mercato. La query riportata è solo d'esempio opportunamente modificata con codice javascript, è possibile dinamicamente modificare i dati inseriti:

```
PREFIX eve:<http://purl.org/NET/c4dm/event.owl#>
PREFIX agreve:<http://www.semanticweb.org/agrievents-ontology#>
PREFIX time:<http://www.w3.org/2006/time#>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX dc:<http://purl.org/dc/elements/1.1/>
PREFIX wgs84_pos:<http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX locn:<http://www.w3.org/ns/locn#>

INSERT DATA {
  agreve:Inizio_Mercato_Palermo a owl:NamedIndividual;
                                a time:Instant;
                                dc:date "2015-12-31T09:30:00+01:00"^^xsd:dateTime
};

INSERT DATA {
  agreve:Fine_Mercato_Palermo a owl:NamedIndividual;
                              a time:Instant;
```



```
        dc:date "2015-12-31T19:30:00+01:00"^^xsd:dateTime
    };
```

```
INSERT DATA {
    agreve:Giorno_1_Mercato_Palermo a owl:NamedIndividual;
        a time:DateTimeInterval;
        time:hasBeginning agreve:Inizio_Mercato_Palermo;
        time:hasEnd agreve:Fine_Mercato_Palermo
};
```

```
INSERT DATA {
    agreve:Mercato_Palermo_Geometry a owl:NamedIndividual;
        a locn:Geometry;
        wgs84_pos:lat "38.108801";
        wgs84_pos:long "13.361886"
};
```

```
INSERT DATA {
    agreve:Mercato_Palermo_Addresss a owl:NamedIndividual;
        a locn:Address;
        rdfs:label "Via del Vespro, 16, 90127, Palermo, Italia";
};
```

```
    locn:fullAddress "Via del Vespro, 16, 90127, Palermo, Italia";
    locn:thoroughfare "Via del Vespro";
    locn:locatorDesignator "16";
    locn:postCode "90127";
    locn:postName "Palermo";
    locn:adminUnitL1 "IT"
};
```

```
INSERT DATA {
    agreve:Mercato_Palermo a owl:NamedIndividual;
    a eve:Event;
    locn:address agreve:Mercato_Palermo_Address;
    locn:geometry agreve:Mercato_Palermo_Geometry;
    eve:time agreve:Giorno_1_Mercato_Palermo
}
```

Con questa semplice query invece l'organizzatore potrà visualizzare tutti gli eventi:

```
PREFIX eve:<http://purl.org/NET/c4dm/event.owl#>
SELECT DISTINCT ?event {
  ?event a eve:Event
} order by ?event
```

Con questa query è possibile visualizzare gli eventi a cui partecipa un agricoltore:

```
PREFIX eve:<http://purl.org/NET/c4dm/event.owl#>
PREFIX agreve:<http://www.semanticweb.org/agrievents-ontology#>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT DISTINCT ?event {
  ?event a eve:Event .
  ?person a foaf:Person .
  ?event eve:agent agreve:Mario_Rossi
} order by ?event
```

Tramite questa query si lega l'agricoltore all'evento

```
PREFIX eve:<http://purl.org/NET/c4dm/event.owl#>
PREFIX agreve:<http://www.semanticweb.org/agrievents-ontology#>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
```

```

INSERT DATA {
    agreve:Mario_Rossi a owl:NamedIndividual;
                        a foaf:Person;
}

INSERT DATA {
    agreve:Mercato_Catania eve:agent agreve:Mario_Rossi
}

```

Tramite la seguente query un'agricoltore può inserire offerte di frutta, la query anche in questo caso è solo un esempio:

```

PREFIX eve:<http://purl.org/NET/c4dm/event.owl#>
PREFIX gr:<http://purl.org/goodrelations/v1#>
PREFIX agreve:<http://www.semanticweb.org/agrievents-ontology#>
PREFIX dbpedia:<http://dbpedia.org/resource/>

INSERT DATA {
    agreve:Vendita_Rossi_Mele a owl:NamedIndividual ;
                              a gr:Offering ;
    gr:includes dbpedia:Apple .
    agreve:Mario_Rossi gr:offers agreve:Vendita_Rossi_Mele .
    agreve:Mercato_Catania eve:product agreve:Vendita_Rossi_Mele
}

```