Virtual Knowledge Graph Generation from Heterogeneous Data Sources

David Chaves-Fraga, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain
Freddy Priyatna, OEG-UPM
Oscar Corcho, OEG-UPM

schema:email = lower(substr({name},1,1) || {surname} || '@fi.upm.es')

Contact:
mdchaves@fi.upm.es
@mdchavesf

Date: 24-25/10/2018
Venue: imec/Ghent University
PhD Student and Researcher at OEG-UPM since 2016 (Data Integration team):

- PhD Thesis (2016-2020): Virtual Knowledge Graph Generation from heterogeneous resources

Interests:

- OBDA
- Heterogeneous data
- SPARQL
- Federated queries
- Data Integration
- Public Transport
- Linked Connections
- R2RML- RML
- Virtualization - Access

@dchavesf
dchaves@fi.upm.es
dchaves.oeg-upm.net
dachafra
OBDA... Ontology Based Data Access

Focused on optimizing the generated SQL query to improve the performance
But we are working on… Semantic **WEB**

How is the data exposed on the Web?

<table>
<thead>
<tr>
<th>Formats</th>
<th>Formato</th>
<th>Formatos</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CSV</strong> (114629)</td>
<td>CSV (10581)</td>
<td>CSV (367)</td>
</tr>
<tr>
<td>TXT (80014)</td>
<td>XLS (7474)</td>
<td>XML (130)</td>
</tr>
<tr>
<td>JSON (50676)</td>
<td>JSON (7234)</td>
<td>XLS (128)</td>
</tr>
<tr>
<td>ZIP (50070)</td>
<td>HTML (6245)</td>
<td>XLSX (88)</td>
</tr>
<tr>
<td>HTML (45706)</td>
<td>PDF (3909)</td>
<td>WMS (29)</td>
</tr>
<tr>
<td>GMZ (44712)</td>
<td>XML-APP (2721)</td>
<td>RDF (21)</td>
</tr>
<tr>
<td>PDF (34770)</td>
<td>XLSX (2649)</td>
<td>GeoJSON (7)</td>
</tr>
<tr>
<td>XLS (26356)</td>
<td>PC-Axis (2490)</td>
<td>JSON (7)</td>
</tr>
<tr>
<td>SHP (19778)</td>
<td>XML (1951)</td>
<td>prj (7)</td>
</tr>
<tr>
<td>XML (19311)</td>
<td>ASCII (1909)</td>
<td>SHP (7)</td>
</tr>
</tbody>
</table>
OBDA... Ontology Based Data Access

SPARQL-to-SQL

CSV
OBDA... Ontology Based Data Access

SPARQL-to-SQL

CSV
CSV → CSV
CSV
CSV → CSV
CSV
Multiple CSV files with relations among them:

1. Joins are not explicit
2. Constraints are not defined explicitly in the CSVs (PK, FKs)
3. The data may not be in the desirable format (e.g. dates)
4. CSVs are not in 3NF:
   a. PK may be repeated
   b. FKs may not be explicited
   c. FKs could not have a 1:1 cardinality
   d. Lists in column

R2RML is not enough for dealing with CSV(s) in an OBDA approach
Let’s give an example...
## LD Generation from GTFS to LinkedGTFS (in hours)

<table>
<thead>
<tr>
<th>Dataset (size mg)</th>
<th>Morph-R2RML (hours)</th>
<th>RML-Mapper (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1 (2.3)</td>
<td>0.004</td>
<td>3.739</td>
</tr>
<tr>
<td>D2 (2.6)</td>
<td>0.026</td>
<td>2.587</td>
</tr>
<tr>
<td>D3 (2.9)</td>
<td>0.068</td>
<td>0.778</td>
</tr>
<tr>
<td>D4 (3.4)</td>
<td>0.118</td>
<td>7.026</td>
</tr>
<tr>
<td>D5 (4.2)</td>
<td>0.115</td>
<td>7.026</td>
</tr>
<tr>
<td>D6 (4.7)</td>
<td>0.217</td>
<td>12.218</td>
</tr>
<tr>
<td>D7 (31)</td>
<td>1.153</td>
<td>151.541</td>
</tr>
<tr>
<td>D8 (96)</td>
<td>12.496</td>
<td>&gt;160</td>
</tr>
</tbody>
</table>
Our mission as researchers is to provide solutions for:

- **Generate Linked Data when:**
  - Quality is important
  - The data is static

- **Access to data using a graph query language when:**
  - Data is volatile
  - Performance is relevant
  - Underlying query engine for translation exists

**Common point: The mapping language!**
How to provide access/generation to heterogeneous data exposed on the web with relations among them using semantic technologies?

How can we extend standard Mapping Languages maintaining their semantics for using OBDA engines or LD generators?
Core Mapping Language + extensions

- RML + FnO
- RML
- RMLC
- R2RML-F
- xR2RML
- Core Mapping Language
<table>
<thead>
<tr>
<th>Feature</th>
<th>R2RML</th>
<th>RML</th>
<th>RML-C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data format</td>
<td>RBD</td>
<td>JSON, CSV, XML</td>
<td>CSV</td>
</tr>
<tr>
<td>Materialization</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Virtualization</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Functions</td>
<td>No</td>
<td>Yes (FnO)</td>
<td>Yes (SQL Functions)</td>
</tr>
<tr>
<td>Specification</td>
<td>Yes</td>
<td>Partially? (FnO+RML?)</td>
<td>Partially</td>
</tr>
</tbody>
</table>
“Virtual Statistics Knowledge Graph Generation from CSV files” D. Chaves-Fraga, F. Priyatna, I. Santana-Perez and O. Corcho at SemStats Workshop co-located with ISWC18 (Best Paper Award)

“SATET: Providing access to multiple CSV on the Web using OBDA” D. Chaves-Fraga and O. Corcho (on-going work)
Virtual Statistics
Knowledge Graph
Generation from CSV files

David Chaves-Fraga, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain
Freddy Priyatna, OEG-UPM
Idafen Perez-Santana, OEG-UPM
Oscar Corcho, OEG-UPM

dchaves@fi.upm.es
@dchavesf
08/10/2018
ISWC18-SemStats
The size of the R2RML mapping depends on the number of columns in the CSV.

Difficulty of maintenance and creation
Two variables for identifying independently each TriplesMap and provide access to the CSV data: \{\$column\}, \{\$alias\}
RMLC-Iterator Example

```
<TriplesMap2016{$column}>
rr:logicalTable {
  rr:tableName "\"2016-P21\"",
  rr:columns ["Jan","Oct","Dec"],
  rr:dictionary {
    "Jan": "January",
    "Oct": "October",
    "Dec": "December"
  }
};

rr:subjectMap {
  a rr:Subject;
  rr:template "http://ex.org/2016{$column}";
  rr:termType rr:IRI;
  rr:subject qb:Observation;
};

rr:predicateObjectMap[
  rr:predicate sltsv:month;
  rr:objectMap[
    rr:termType rr:IRI;
    rr:constant "http://reference.data.gov.uk/def/intervals/{$alias}";
  ];
];

rr:predicateObjectMap[
  rr:predicate sltsv:numberOfArrivals;
  rr:objectMap[
    rr:termType rr:Literal;
    rr:column {$alias};
    rr:datatype xsd:integer;
  ];
];
```

Accessed columns
Dictionary with alias
Reference to columns
1 TriplesMap for 12 Months
Reference to alias
Output and Results

Outputs:

- RMLC-Iterator for transforming the mappings to R2RML
- Morph-RDB as OBDA engine for the query translation

Results:

<table>
<thead>
<tr>
<th>Features</th>
<th>R2RML</th>
<th>RMLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Lines</td>
<td>~700</td>
<td>74</td>
</tr>
<tr>
<td>#TriplesMaps / #SubjectMaps</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>#PredicateObjectMaps</td>
<td>60</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Features</th>
<th>R2RML</th>
<th>RMLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Lines</td>
<td>&gt;2800</td>
<td>&lt;70</td>
</tr>
<tr>
<td>#TriplesMaps / #SubjectMaps</td>
<td>&gt;40</td>
<td>1</td>
</tr>
<tr>
<td>#PredicateObjectMaps</td>
<td>&gt;170</td>
<td>4</td>
</tr>
</tbody>
</table>
SATET: Providing access to multiple CSV on the Web using OBDA

David Chaves-Fraga, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain
Freddy Priyatna, OEG-UPM
Oscar Corcho, OEG-UPM

dchaves@fi.upm.es
@dchavesf

???

???
Semantic Interoperability in CSV files - Virtualizing

CSV + R2RML = RESULT SET

SPARQL + SQL = RESULT SET
Multiple CSV files with relations
SATET: Semantic Access for heterogeneous Tabular data

- RMLC: Extension of R2RML for including SQL functions
  - Discover implicit joins among CSV files
  - Transforming CSV columns to RDF objects
- Generation of an enriched database schema using the mapping info (optimization)
- Semantic preservation of R2RML
Join in R2RML and RML

Diagram:
- `rr:PredicateObjectMap` to `rr:RefObjectMap`
- `rr:RefObjectMap` to `rr:TriplesMap`
- `rr:TriplesMap` to `rr:JoinCondition`
- `rr:JoinCondition` to `column name`
- `rr:JoinCondition` to `column name`
Discovering implicit joins between CSV files

Relational Database

<table>
<thead>
<tr>
<th>id, name, surname, birthdate, location</th>
<th>id, job</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, david, chaves-fraga, 27-11-1993, SDC</td>
<td>1, phd_student</td>
</tr>
</tbody>
</table>

CSV files

<table>
<thead>
<tr>
<th>name, surname, birthdate, location</th>
<th>full_name, job</th>
</tr>
</thead>
<tbody>
<tr>
<td>david, chaves_fraga, 27111993, SDC</td>
<td>&quot;David Chaves Fraga&quot;, &quot;phd_student&quot;</td>
</tr>
</tbody>
</table>
RMLC: RDF Mapping Language extension for heterogeneous CSV files

The functions are SQL basic transformation functions
Table 1

name, surname, birthdate, location
david, chaves_fraga, 27111993, SDC

Table 2

full name, job
"David Chaves Fraga", "phd_student"

SELECT ?name ?birthday ?job WHERE {
}
SELECT ?name ?birthday ?job
WHERE {
}

SELECT name, birthday, table2.job FROM table1
INNER JOIN table2 ON
CONCAT(table1.name, ',', REPLACE(table1.surname, '_', '')) = LOWER(table2.full_name)
Example with GraphQL

```json
{
  "data": {
    "listSocialMediaPosting": [
      {
        "identifier": "http://ex.org/1",
        "comment": "Hallo Dunia@id",
        "author": {
          "identifier": "http://ex.org/Person/1",
          "email": "fpriyatna@fi.upm.es",
          "familyName": "Priyatna",
          "givenName": "Freddy",
          "name": "Freddy Priyatna",
          "telephone": "8141"
        }
      },
      {
        "identifier": "http://ex.org/2",
        "comment": "Hola Mundo@es",
        "author": {
          "identifier": "http://ex.org/Person/2",
          "email": "dchaves@fi.upm.es",
          "familyName": "Chaves",
          "givenName": "David",
          "name": "David Chaves",
          "telephone": "9063"
        }
      },
      {
        "identifier": "http://ex.org/3",
        "comment": "Hello World"
      }
    ]
  }
}

SELECT
  "listSocial"."id" AS "id",
  'http://ex.org/' || "listSocial"."id" || ' ' AS "identifier",
  "listSocial"."mensaje" AS "comment",
  "author"."id" AS "author__id",
  'http://ex.org/Person/' || "author"."id" || ' ' AS "author__identifier",
  lower(substr("author"."nombre",1,1)) || "author"."apellido" || '@fi.upm.es') AS "author__email",
  "author"."apellido" AS "author__apellido",
  "author"."nombre" AS "author__nombre",
  ' ' || "author"."nombre" || ' ' || "author"."apellido" || ' ' AS "author__name",
  "author"."telephone" AS "author__telephone"
FROM comentarios "listSocial"
LEFT JOIN personas "author" ON "listSocial"."usuario" = lower(substr("author"."nombre",1,1)) || "author"."apellido"
```

Chaves-Fraga / Virtual Knowledge Graph Generation
<TriplesMap1>
...
rr:predicateObjectMap[
    rr:predicate ex:shortName;
    rr:objectMap [
        rr:datatype xsd:string;
        rmlc:functions "REPLACE(SUBSTRING(LOWER{FULL_NAME},1,5),' ','−')";
    ]; ];
rr:predicateObjectMap[
    rr:predicate ex:yearofBirthday;
    rr:objectMap [
        rmlc:functions "YEAR({birthday})"; ]; ];
Enriched database schema

- Primary Keys
- Foreign Keys
- Datatypes

CSV → CSV + RMLC = SATET

++
Alignment with R2RML and OBDA engines

- RMLC maintains the semantics of R2RML
- It’s aligned with R2RML:
  - ObjectMaps with Functions → new column in the table with the name of the predicate
  - Joins with Functions → new columns in the tables
  - SATET transforms RMLC to R2RML
- SATET can be introduced on the top of state-of-art OBDA engines (morph/ontop) for using their optimizations to efficiently access to CSV files
SELECT ?name ?birthday ?job
WHERE {
}

SELECT name, birthday, table2.job FROM table1
INNER JOIN table2 ON table1.fullName = table2.fullName
GTFS to RDF materialization

- RML-Mapper
- Morph-rdb
- RMLC

GTFS datasets
SATET: Semantic Access for heterogeneous Tabular data

Main Contributions:
- Discover implicit joins
- Apply transformation functions to individual columns
- Enriched database schema from mapping information
- Semantic preservation of R2RML

Future Work:
- Alignment with FnO → full specification (Possible collaboration)
- Alignment with RML (without FnO) for LD Generation from RDB/CSV
- Optimizations over generated SQL queries
- Query answering over SATET
- Applying to transport domain for linking potential datasets during a route planning creation
- Define the core for the mapping languages of SW (Possible collaboration)
Virtual Knowledge Graph Generation from heterogeneous data sources

David Chaves-Fraga, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain
Freddy Priyatna, OEG-UPM
Oscar Corcho, OEG-UPM

dchaves@fi.upm.es
@dchavesf

24-25/10/2018
@imec/Ghent University