Mapping Spreadsheets to RDF – Supporting Excel in RML

Markus Schröder, Christian Jilek, Andreas Dengel

Second International Workshop on Knowledge Graph Construction

06.06.2021
A Spreadsheet
(or sometimes a canvas painted by a data artist)

◆ Spreadsheets
  ▪ well understood
  ▪ easy and fast possibility to enter data
  ▪ complex workbooks
  ▪ multiple sheets
  ▪ cells having rich meta data
    o formats, colors, fonts, styles, borders, etc.
    o arbitrarily arranged
    o can lead to inconsistent and unstructured content

Found at data.gov:
Hutten 2016_CRMO_lichen_moss_liverwort_.xlsx
From Spreadsheet to RDF

- Use input format that is supported
- Advantages of native support
  - Eliminates extraneous conversion efforts: No preprocessing and transformation needed
  - All aspects of a spreadsheet can be exploited
  - Eases mapping rule communication
  - For RML practitioners no extra language to learn
Related Work

Spread2RDF (Ruby syntax)

```ruby
worksheet 'MaterialelementeKlassen',
  name: 'MaterialelementeKlassen',
  start: 10,
  subject: { url: { namespace: PSM.MaterialElement },
  type: RDF:RDFS.Class,
  sub_class_of: PSM.MaterialElement
  }
do
  column :name, predicate: RDFS.label

column :uri
  column :sub_class_of,
    predicate: RDFS.subClassOf,
    object: { from: 'MaterialelementeKlassen' }

column :block :parameter,
    subject: { url: :node, type: PSM.Parameter },
    predicate: PSM.materialParameter,
    statement: :restriction do
  column :name, predicate: PSM.parameterName
  column :description, predicate: PSM.parameterDescription

column :min, predicate: PSM.parameterMinQuantity,
    object: { url: :node, type: QUOT.QuantityValue },
  &quantity_mapping
```

Mapping Master (Manchester syntax)

Class: R*(rdfs:label 'analyte assay')

EquivalentTo:
  (achieves_planned_objective some 'analyte measurement objective')
  (realizes some 'evaluative role' and role_of some
  R*(material_entity)))
  (realizes some 'analyte role' and
  role_of some 'scattered molecular aggregate' and
  (has_unit only
  R*(molecular_entity))))

SubClassOf:
has_scheduled_output some
  (scalar measurement datum and
  (is_quality_measurement of some 'molecular concentration') and
  ('has measurement unit label' some
  R*(measurement_unit_label)))

Sheet2RDF (PEARL syntax)

```pearl
[] a x1:Mapping;
...
@templateGraph :ANamedGraph;
...
@ANamedGraph{
  @x1:"http://example.org/" @URLENCOD(A2 & B2)""@x1:Expr ] a foaf:Person ; foaf:name A2 & B2 -x1:Expr ; foaf:homepage "http://example.org"@x1:Expr ;
```
Approach: Technical Integration in RMLMapper

◆ Extension to the RML Mapper tool

GitHub fork

◆ Record Factory
  - Use parameters to return records
  - Apache POI

◆ Record
  - Interpret references
Conceptual Integration in RML
Cell Location Reference

◆ Cell iteration using ranges (e.g. B2:B5)
  ▪ Instead of row iteration (like in CSV)
◆ Arbitrarily structured tables without any anchor points like column names
  ▪ Relative reference to neighboring cells with parenthesis notation
  ▪ Absolute reference to cells with square brackets

```
rr:_predicateObjectMap [  
a rr:PredicateObjectMap ;  
rr:_predicateMap [  
  a rr:PredicateMap ;  
  rr:constant ex: numberOfPages  
] ;  
rr:objectMap [  
a rr:ObjectMap ;  
  rml:reference "(1,0).valueInt" ;  
  rr:datatype xsd:integer  
] .
```

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Title</td>
<td>Pages</td>
<td>Price</td>
</tr>
<tr>
<td>2</td>
<td>Deducing Supplies</td>
<td>4</td>
<td>$10.40</td>
</tr>
<tr>
<td>3</td>
<td>Structuring Fair-Med</td>
<td>6</td>
<td>$5.23</td>
</tr>
<tr>
<td>4</td>
<td>Utilizing Symbolic</td>
<td>8</td>
<td>$2.50</td>
</tr>
<tr>
<td>5</td>
<td>Colonizing Loyal</td>
<td>3</td>
<td>$11.23</td>
</tr>
</tbody>
</table>

x-offset: -1 0 +1
y-offset: -1 0 +1

(1,0)
Conceptual Integration in RML
Meta Data References

◆ Access cell meta data
- address
- column
- row
- backgroundColor
- foregroundColor
- fontColor
- fontName
- fontSize
- valueNumeric
- valueInt
- valueBoolean
- valueFormula
- valueString
- json
- json
- value

Colonizing Loyal Ethos and Artifice

```
valueString

A5
address

backgroundColor

json

{  
  "address": "A5",
  "column": 0,
  "row": 4,
  "cellType": "string",
  "valueNumeric": 0,
  "valueBoolean": false,
  "valueString": "Colonizing Loyal Ethos and Artifice",
  "valueRichText": "<font face='Arial'>Colonizing (...)",
  (...) } ```

Colonizing Loyal Ethos and Artifice

```
<font face='Arial'>Colonizing Loyal </font>
<b><font face='Arial'>Ethos</font></b>  
<font face='Arial'>and </font>
<i><font face='Arial'>Artifice</font></i>  
```
Experimental Features
Multiple Different Properties in a Cell

◆ Each piece of information corresponds to a different property

◆ Zip together
  ▪ properties
  ▪ returned objects from FnO function

◆ Just a shortcut
  ▪ Usually requires separate predicate object maps

```rml
rr:predicateObjectMap {
  a rr:PredicateObjectMap ;
  ss:zip true ;
  rr:predicateMap {
    a rr:PredicateMap ;
    rr:constant ( ax:numberOfPages ox:price )
  } ;
  rr:objectMap {
    a rri:ObjectMap , fnml:FunctionMap ;
    fnml:functionValue {
      rr:predicateObjectMap {
        rr:predicate fno:executes ;
        rr:object <java:ifRegexReturnGroup> ;
      }
    }
  }
}
```

Result
Experimental Features
Multiple Complex Entities in a Cell

- For example: list of persons having first and last names
- RDF graph in turtle syntax
- New term type ss:Graph
- Graph is added to result

Selected ones are mapped using ss:SelectedObjects
Conclusion

- Implemented Excel support in RML Mapper
- Cell iteration, location, meta data access
- Try it out on our demo page
  - GitHub code

- You may also be interested in
  - Data Sprout – Dataset generation for evaluating KGC

- More related research in our project
  - SensAI

Thank you for your attention. Questions?