Transforming Data from DataPile Structure into RDF

Jiří Dokulil
Faculty of Mathematics and Physics
Charles University in Prague
DataPile

- Data verticalization
- Metadata
RDF

• triples
  – subject – resource being described
  – predicate – property
  – object – value of the property

• metadata
  – stored with data
### DataPile → RDF

<table>
<thead>
<tr>
<th>EntID1</th>
<th>attribute 1</th>
<th>value 11</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntID1</td>
<td>attribute 2</td>
<td>value 12</td>
<td>...</td>
</tr>
<tr>
<td>EntID1</td>
<td>attribute 3</td>
<td>value 13</td>
<td>...</td>
</tr>
<tr>
<td>EntID2</td>
<td>attribute 1</td>
<td>value 21</td>
<td>...</td>
</tr>
<tr>
<td>EntID2</td>
<td>attribute 2</td>
<td>value 23</td>
<td>...</td>
</tr>
<tr>
<td>EntID2</td>
<td>attribute 4</td>
<td>value 24</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

The DataPile contains entities with attributes and values. Each entity (EntID1 and EntID2) has several attributes with corresponding values. These entities and their attributes are then converted into an RDF graph, where each entity is represented as a node and the attributes as edges connecting the nodes. The values associated with the attributes are also represented as literals in the RDF graph.
DataPile → RDF example

| 598472635 | first_name | Jiří | … |
| 598472635 | last_name  | Dokulil | … |
| 598472635 | place_of_birth | 972324584 | … |

_:598472635 pile:first_name “Jiří“
_:598472635 pile:last_name “Dokulil“
_:598472635 pile:place_of_birth _:972324584
Special case: multilingual attributes

• complicated in DataPile
  – 2 entities, hard to work with
  – not only language but cases as well

• RDF has direct support for specifying language of literals
  – RFC 3066 → flexible enough to express language, case, …
Metadata

• separated from data in the DataPile
  – tree, 2 levels
    • entities
    • attributes
  – data types
    • string, number, timestamp
    • ID – reference to other entity – typed reference

• everything is a triple in RDF
  – easy to represent the metadata tree from DataPile
  – data types taken from XSD (and simplified)
    • enough to cover types in DataPile
  – object of a triple can be anything → we need more constraints → RDFS
RDFS – RDF Schema

• predicates have to be URIs (resources) → attributes from DataPile must be transformed to RDF
• transforming entities is easy
• entities serve as domains of predicates
• ranges of predicates can be either specific data types or instances of other entities → RDFS can provide same level of type checking as DataPile
• DataPile does not support is-a hierarchies while RDFS can define subclasses and subproperties
Reification – making statements about statements

- resource 1
  - property 1
  - property 2
- resource 2
  - property 1
  - property 2
  - property 3
- literal 1
- literal 3
- literal 2
### Reification in DataPile

<table>
<thead>
<tr>
<th>EntID1</th>
<th>attribute 1</th>
<th>value 11</th>
<th>valid_from 1</th>
<th>valid_to 1</th>
<th>modified_by 1</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntID1</td>
<td>attribute 2</td>
<td>value 12</td>
<td>.valid_from 2</td>
<td>valid_to 2</td>
<td>modified_by 2</td>
<td>...</td>
</tr>
<tr>
<td>EntID1</td>
<td>attribute 3</td>
<td>value 13</td>
<td>.valid_from 3</td>
<td>valid_to 3</td>
<td>modified_by 3</td>
<td>...</td>
</tr>
<tr>
<td>EntID2</td>
<td>attribute 1</td>
<td>value 21</td>
<td>.valid_from 4</td>
<td>valid_to 4</td>
<td>modified_by 4</td>
<td>...</td>
</tr>
<tr>
<td>EntID2</td>
<td>attribute 3</td>
<td>value 23</td>
<td>.valid_from 5</td>
<td>valid_to 5</td>
<td>modified_by 5</td>
<td>...</td>
</tr>
<tr>
<td>EntID2</td>
<td>attribute 4</td>
<td>value 24</td>
<td>.valid_from 6</td>
<td>valid_to 6</td>
<td>modified_by 6</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- special case of reification with fixed set of predicates
Reification in RDF

_:568421369754123695 mt:person__name "John Smith" .
_:r65413 rdf:type rdf:Statement .
_:r65413 rdf:subject _:568421369754123695 .
_:r65413 rdf:predicate mt:person__name .
_:r65413 rdf:object "John Smith" .
_:r65413 mt:valid_from "20050703T15:21:49" .
_:r65413 mt:valid_to "20050821T09:35:12" .
Conclusion

• Basic transformation is easy
• Multilingual attributes can be expressed better in RDF
• No real reification in RDF → transforming validity period is not nice