<xs:complexType name="CategoryType">
<xs:sequence>
<xs:element name="description" type="xs:string" />
<xs:element name="category" type="CategoryType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="books">
<xs:sequence>
<xs:element name="book" type="BookType"
minOccurs="0" maxOccurs="unbounded"/>

Linked Data Fusion with Conflicts

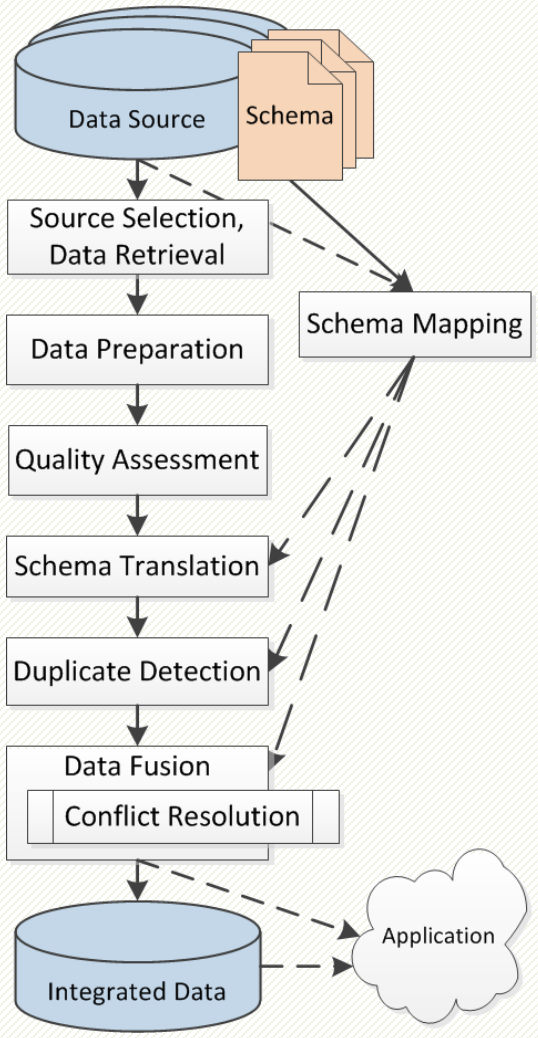
Jan Michelfeit
michelfeit@ksi.mff.cuni.cz
Matematicko-fyzikální fakulta Univerzity Karlovy

Contents

1. **Linked Data Fusion**
2. A fusion & conflict resolution algorithm
3. An (F-)quality assessment algorithm
4. New frontiers
   - Dependent properties
   - Dependent resources
   - Scaling up
5. Future work & summary

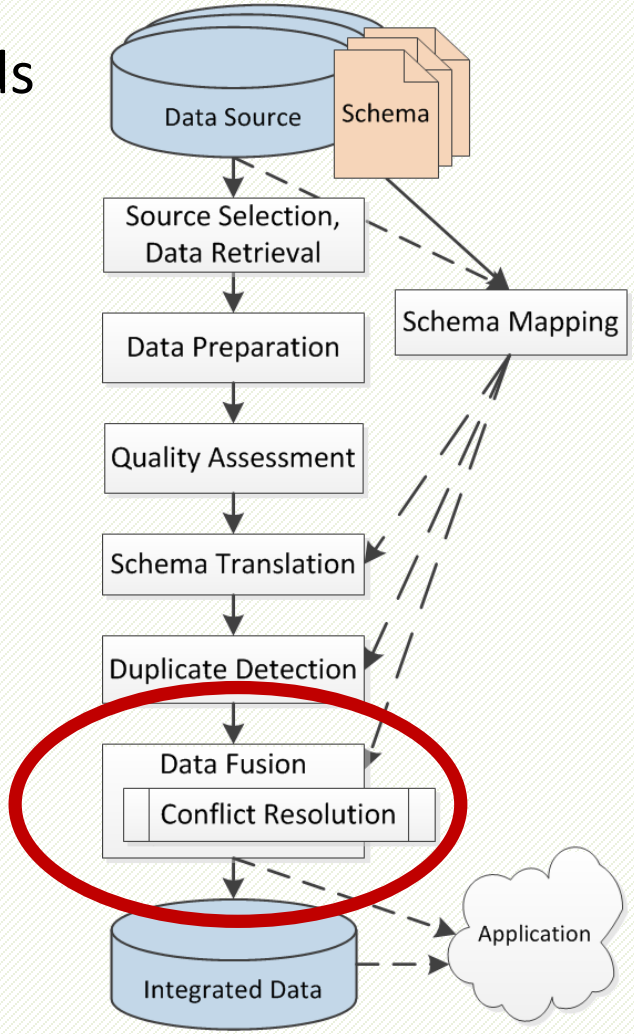
Data Integration

- Multiple sources
  => unified view on data
- Problems
  - Heterogeneity
  - Conflicts
  - Cleansing of incorrect
    or otherwise flawed data
  - Target schema, schema translation



Data Fusion

- Data Fusion – "fusing multiple records representing the same real-world object into a single, consistent, and clean representation"
- Linked Data Fusion
- Conflicts
  - Identity conflicts
  - Schema conflicts
  - Data conflicts

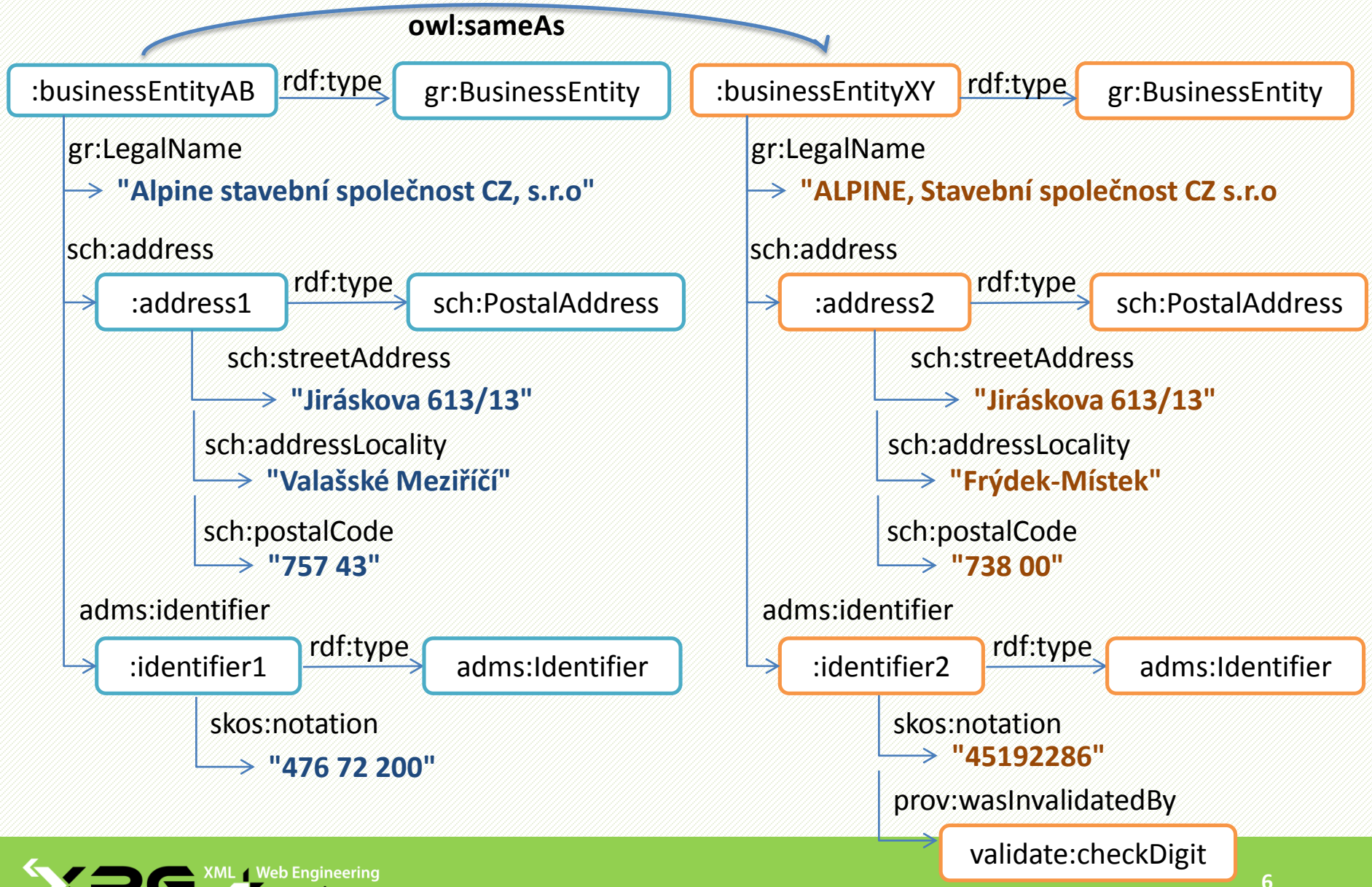


Example – Freebase & DBPedia

```
<http://rdf.freebase.com/ns/en.berlin> rdfs:label  "Berlin"@en ;
   rdf:type       ns:citytown,
   rdf:type       ns:location ;
   rdf:type       ns:travel_destination ;
   ns:area        "891.85"^^xsd:float ;
   ns:latitude    "52.5233"^^xsd:float ;
   ns:longitude   "13.4127"^^xsd:float ;
   ns:population  [
      ns:dated_integer.number  "3442675";
      ns:dated_integer.year    "2009"^^xsd:datetime ] .

<http://dbpedia.org/resource/Berlin> rdfs:label  "Berlin ;
   rdf:type                 dbo:Place ;
   rdf:type                 umbel:City ;
?  dbo:areaTotal            "891850000"^^xsd:double ;
   geo:lat                  "52.500557"^^xsd:float ;
   geo:long                 "13.398889"^^xsd:float ;
   dbo:populationTotal      "3538652" ;
   dbo:populationAsOf       "2012-10-31"^^xsd:date .
```

Example – Věstník veřejných zakázek



ODCS-FusionTool

- Linked Data fusion tool with provenance tracking and quality assessment

- Originally ODCleanStore component
  -> Standalone command-line tool
  -> UnifiedViews DPU

Contents

1. Linked Data Fusion
2. **A fusion & conflict resolution algorithm**
3. An (F-)quality assessment algorithm
4. New frontiers
   - Dependent properties
   - Dependent resources
   - Scaling up
5. Future work & summary

A Data Fusion & Conflict Resolution algorithm

- RDF data fusion with provenance tracking and quality assessment
- … a few definitions
  - Triple, quad
  - Object conflict cluster
    - Set of quads sharing the same subject and property
  - Conflict resolution function f(CC, M)
    - Accepts conflict cluster and metadata
    - Produces resolved quads

```
<http://rdf.freebase.com/ns/en.berlin>
  ns:area "891.85".
<http://dbpedia.org/resource/Berlin>
  dbo:areaTotal "891850000".
```

Conflict resolution functions

- ALL
- ANY
- BEST
- Latest
- BestSource
- MostSpecific
- …

Deciding

- AVG
- SUM
- Concat
- TokenUnion
- …

Mediating

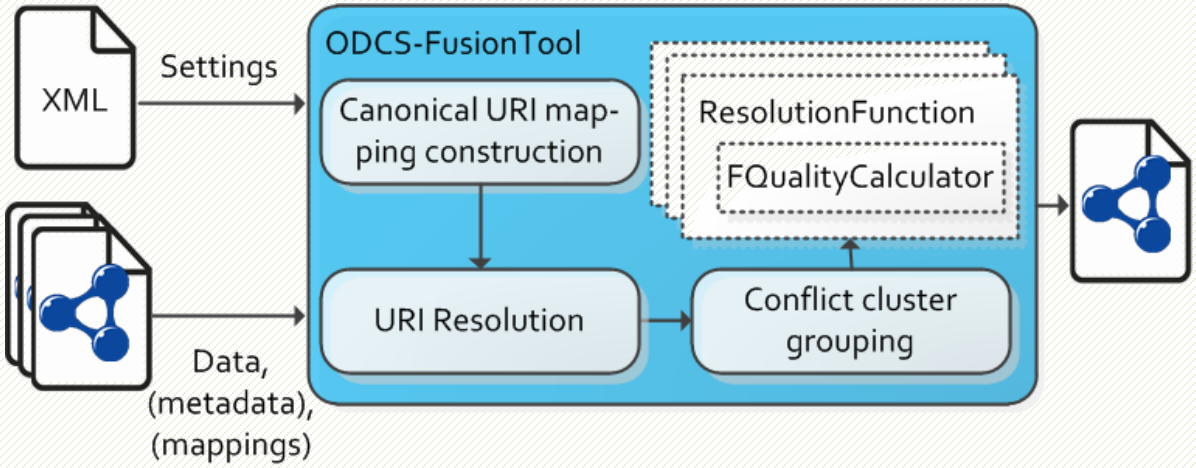
A Data Fusion algorithm – I/O

- Input:
  - Quads to be resolved
  - Metadata – quads (e.g. odcs:score)
  - Mappings - owl:sameAs links
  - Settings – e.g. default/per-property resolution function, multiplicity
- Output:
  - Set of *resolved quads*, e.g. (dbpedia:Belin geo:long "13.391" ng:1, {http://dbpedia.org, http://rdf.freebase.com}, 0.71)

A Data Fusion algorithm

1. Build *canonical URI* mapping from sameAs links
2. Replace URIs with their canonical URI
3. Remove duplicates
4. Group input quads into conflict clusters

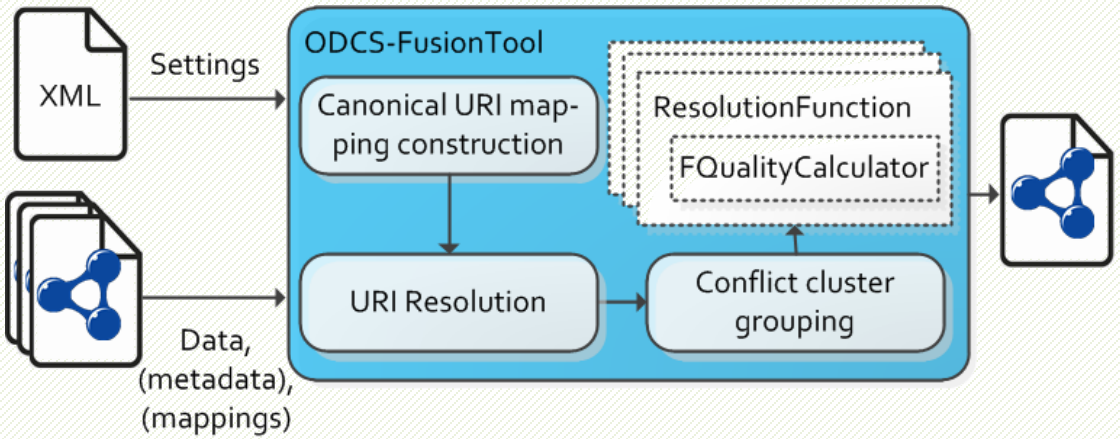
…



A Data Fusion algorithm

...

5. For each conflict cluster $CC_{s,p}$:
   a) Choose resolution function
   b) Apply resolution function – resolve conflicts, track provenance, calculate quality
   c) Add function's output to the result



Contents

1. Linked Data Fusion
2. A fusion & conflict resolution algorithm
3. **An (F-)quality assessment algorithm**
4. New frontiers
   - Dependent properties
   - Dependent resources
   - Scaling up
5. Future work & summary

F-quality

- *F-quality score* is a number from [0,1] expressing the quality of a value after data fusion with respect to other conflicting values, provenance and quality-related metadata.
- Introduced to distinguish from source data quality *(cf. LDIF Sieve)*
- Purpose
  - Deciding factor for conflict resolution
  - Lead for data consumers
  - Detection of low-quality data

A Conflict-based Quality Assessment algorithm

- Based on three factors
  - Quality score of data sources
  - Data conflicts
  - Confirmation of values by multiple sources

A Conflict-based Quality Assessment algorithm

- F-quality of value **v**:
  q(**v**, Sources, ConflictCluster, Metadata)

- Factor 1:
  - $q = \bar{s}(Sources)$ - avg/max of source quality scores
- Factor 2 (only when multiplicity = 1):
  - $q = q \cdot \left(1 - \frac{\sum_{i=1}^{n} s(g_i) \cdot d(\boldsymbol{v}, o_i)}{\sum_{i=1}^{n} s(g_i)}\right)$
  - Distances of other conflicting values weighted by their source quality.

A Conflict-based Quality Assessment algorithm

- q(**v**, Sources, ConflictCluster, Metadata)
  - F-quality of value **v**

- Factor 3 (optional):
  - $q = q + (1 - q) \cdot \min\left(1, \frac{\sum_{g_i \in support} s(g_i) - \max_{g_i \in support} s(g_i)}{AgreeCoefficient}\right)$
  - Increase quality when multiple sources agree on exactly the same value

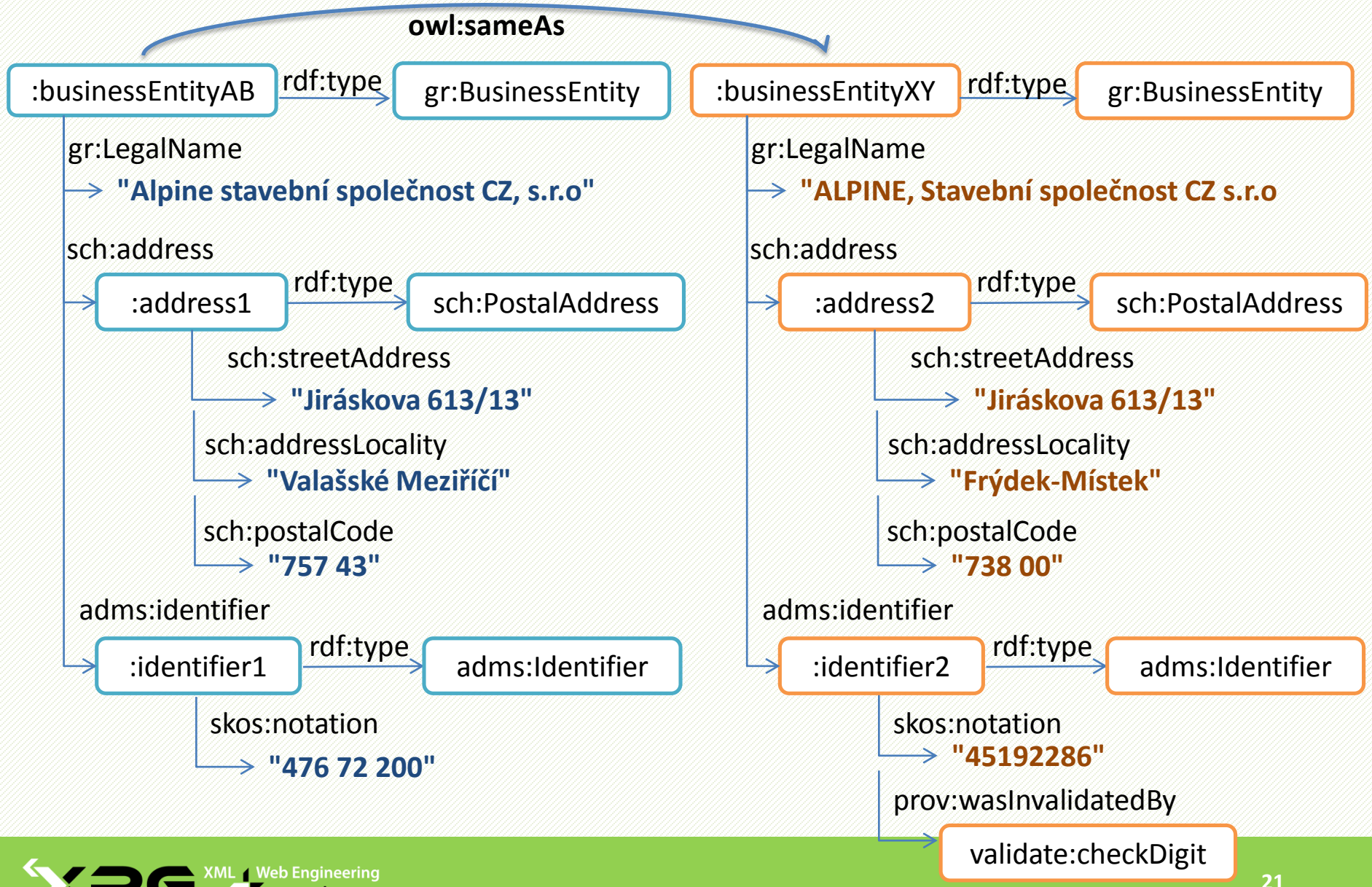
Example of Fused Result

Property	Value	F-Quality	Sources
rdfs:label	"Berlin"	0.75992	DBPedia, Freebase, Geonames
	"City_of_Berlin"	0.27447	Freebase
	"Berlin (Germany)"	0.22126	NYT
geo:lat	"52.5006"	0.72418	DBPedia
	"52.5167"	0.64381	NYT
	"52.5233"	0.64380	Freebase
	"52.52437"	0.64380	Geonames
	"13.4126"	0.15610	Err
geo:long	"13.3989"	0.89957	DBPedia
	"13.4"	0.79965	NYT
	"13.41053"	0.79963	Geonames
	"13.4127"	0.79956	Freebase
dbprop:web	<http://www.berlin.de...php>	0.37739	DBPedia,Freebase
	<http://berlin.unlike.net/>	0.11793	DBPedia
	<http://www.berlin.de>	0.09275	Freebase
	...		
rdf:type	schema:City	0.92000	DBPedia,Freebase
	schema:Place	0.90000	DBPedia
	geonames:Feature	0.80000	Geonames
	...		

Contents

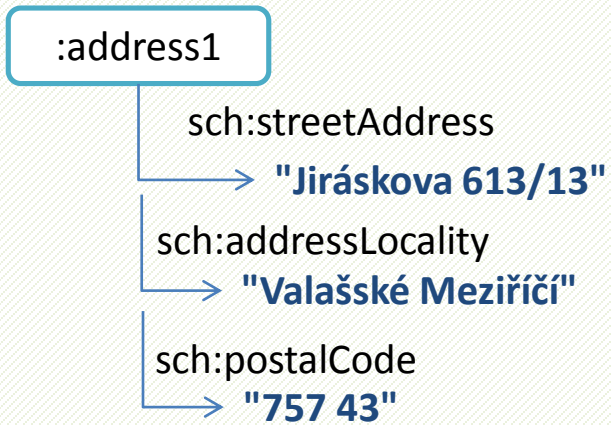
1. Linked Data Fusion
2. A fusion & conflict resolution algorithm
3. An (F-)quality assessment algorithm
4. **New frontiers**
   - **Dependent properties**
   - **Dependent resources**
   - **Scaling up**
5. Future work & summary

Example – Věstník veřejných zakázek



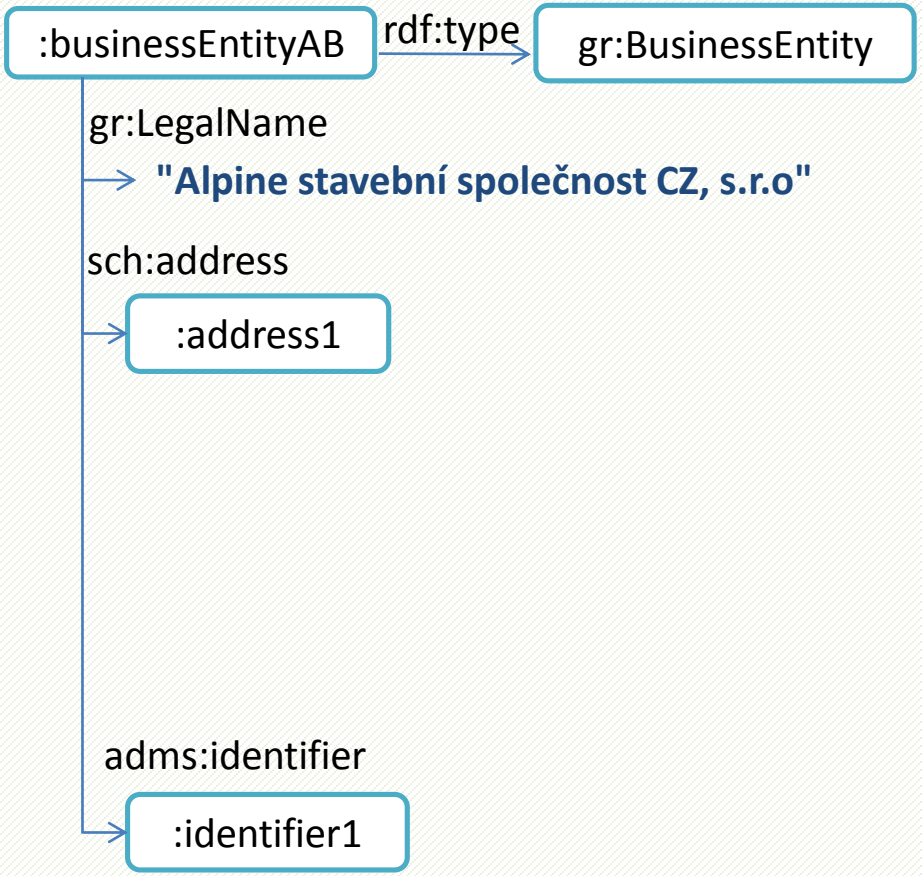
New Features – Dependent Property Groups

- Always keep values for the same subject *before mapping* (and graph ?) together
  - E.g. city & postal code
  - Choose the subject which yields the highest quality
- Open Questions
  - How to calculate quality?
  - How to treat missing values?
  - How to treat manyvalued properties?



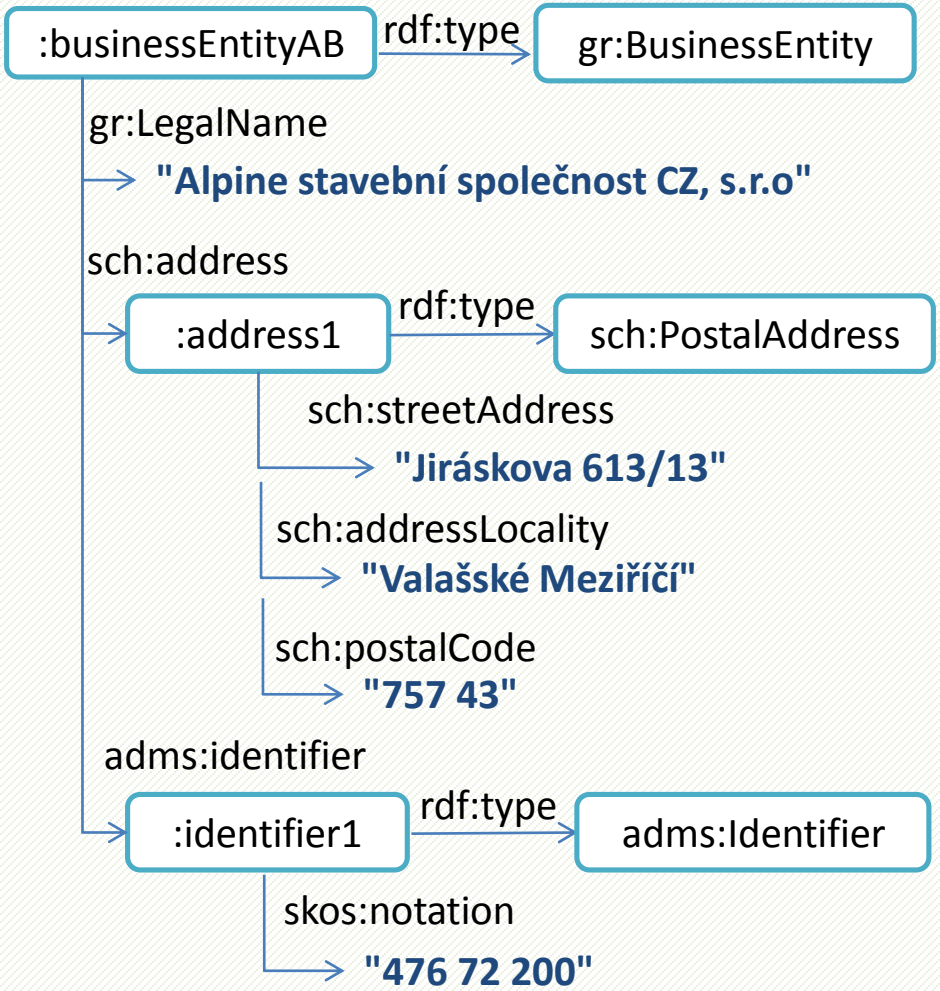
New Features – Dependent Resources

- *Before*:
  resource description
  = triples with the
  same subject



New Features – Dependent Resources

- *Now*:
  include
  dependent resources



New Features – Dependent Resources

- ✓ Able to choose the best dependent resource
- ✓ Able to merge e.g. multiple addresses without explicit sameAs links
- ✓ Quality propagates to root resource

- Open Questions
  - What to include? How can the user define it?
  - How to retrieve efficiently?
  - Can dependent resource be independent?
  - What about cycles?
  - What URI to choose for dependent resources? (Locality of mapping)

New Features – Large Data

- *Before*: resolve query result
- *Now*: Fuse whole data sets

- Open Questions
  - How to connect fused result to other datasets?
    - How to select what should be fused?
  - Orphan (dependent) resources?
  - Efficient implementation?

New Features – Large Data

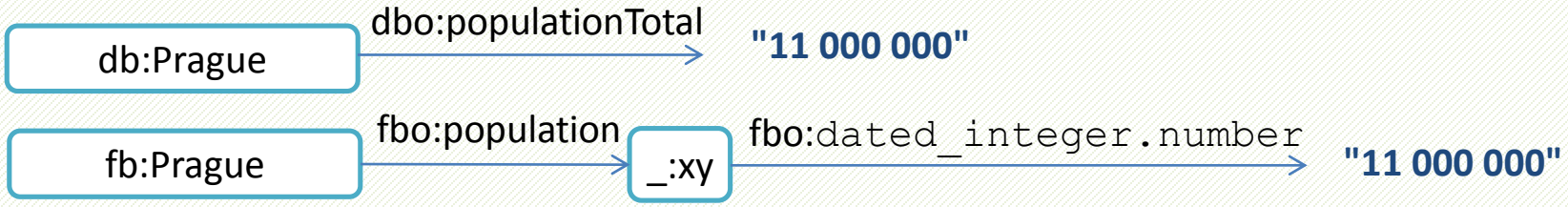
- Main bottleneck – data retrieval
- Data do not fit in memory
- Solution:
  - Download & process data locally
  - Use external sort for conflict clustering
  - Devise data structures/file formats to efficiently retrieve *resource descriptions.*
    - Currently
      - *N-Tuples format*
      - Only dependent resources in depth 1
      - Requires 3 sorts and 1 join
      - data file c(S) S P O G,
      - index files c(D) c(S) and c(S) D P O G for dependent resources

Contents

1. Linked Data Fusion
2. A fusion & conflict resolution algorithm
3. An (F-)quality assessment algorithm
4. New frontiers
   - Dependent properties
   - Dependent resources
   - Scaling up
5. **Future work & summary**

More open questions…

- More complex mappings



- Visualization, evaluation of result
- Quality calculation
  - Malicious sameAs detection
  - Configuration
  - Consider graph structure and similarity
  - Resolution function chaining
  - Integration with SPARQL, …

Summary

- ODCS-FusionTool
  - Identity, schema and data conflict resolution
  - Conflict-based (F-)quality assessment
- New research
  - Dependent properties, dependent resources
  - Scaling up
- Future work
  - A lot…
  - **Anyone would like to join?**

Thank you for your attention

… any questions?

Backup slides

Example – Věstník veřejných zakázek