




# Deriving an Emergent Relational Schema from RDF Data

**Minh-Duc Pham**
*duc@cwi.nl*

Linnea Passing
*linnea.passing@tum.de*

Orri Erling
*oerling@openlinksw.com*

Peter Boncz
*boncz@cwi.nl*


# Main Problems in RDF Data Management [1]

- Bad query plans
- Low storage locality
- Lack of user schema insight

**RDF de-emphasizes the need for a schema and the notion of structure in the data**

Emergent Schema

[1] Minh-Duc Pham, Boncz P.A., " Self-organizing Structured RDF in MonetDB ," PhD Symposium, ICDE, 2013

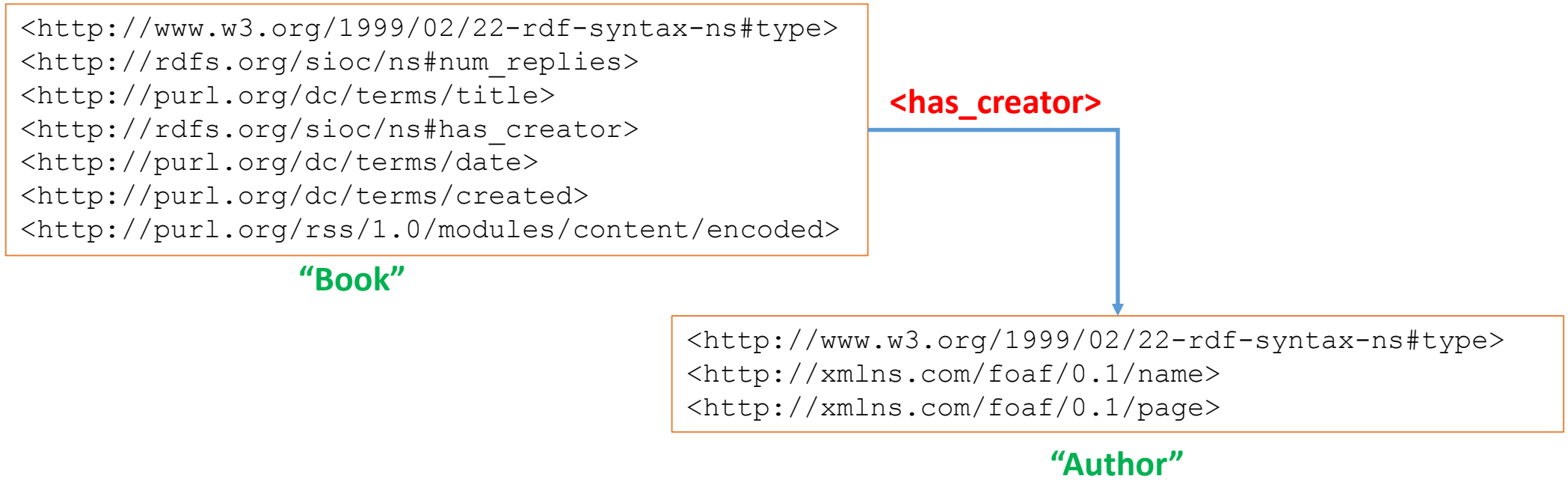
# Recovering the Emergent Schema of RDF data

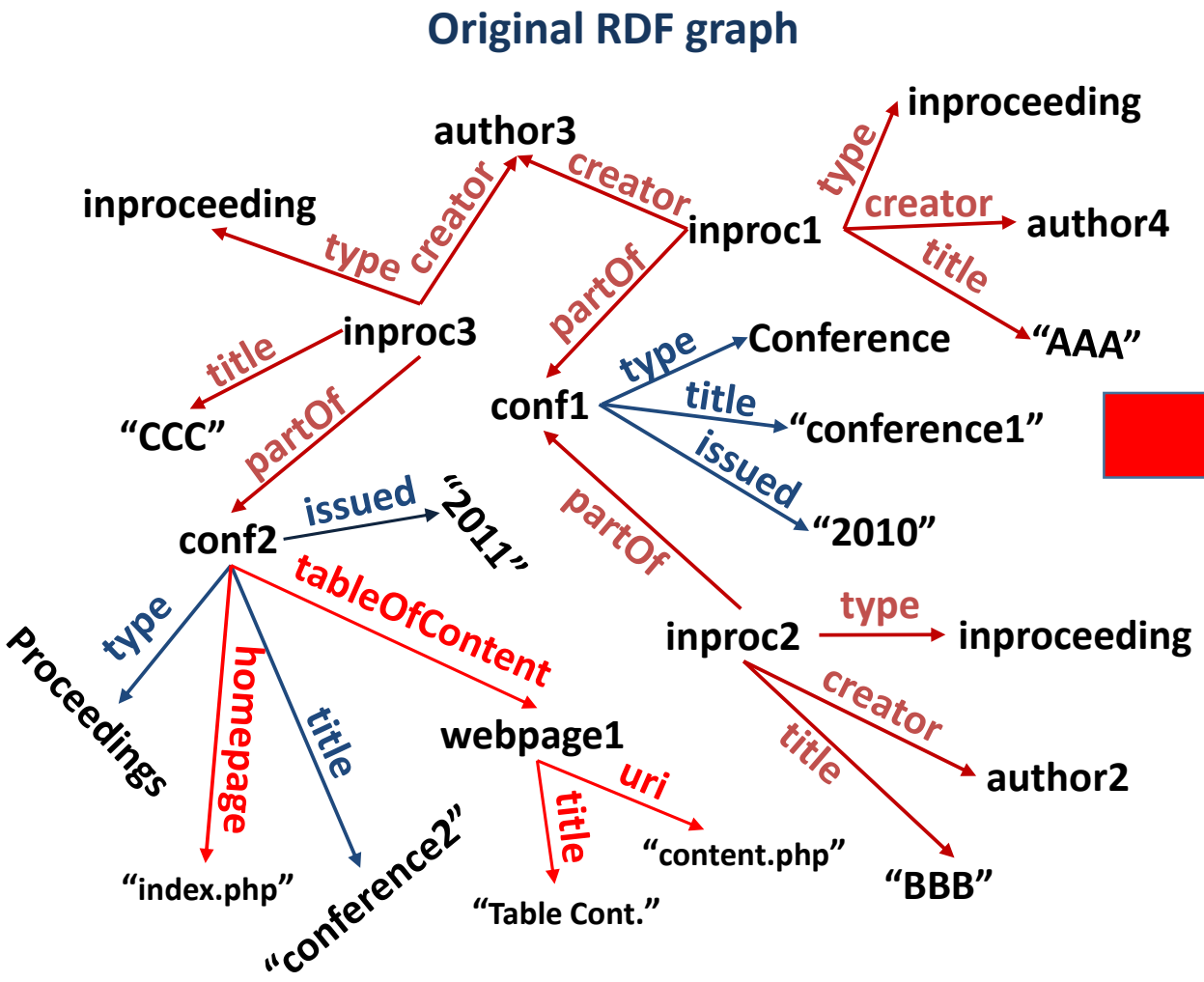
Emergent schema = “rough” schema to which the majority of triples conforms

Recognize:

- **Classes** (Characteristic Sets - CS’s ) – recognize “classes” of often co-occurring properties
- **Relationships** (CS) – recognize often-occurring references between such classes

+ give logical names to these



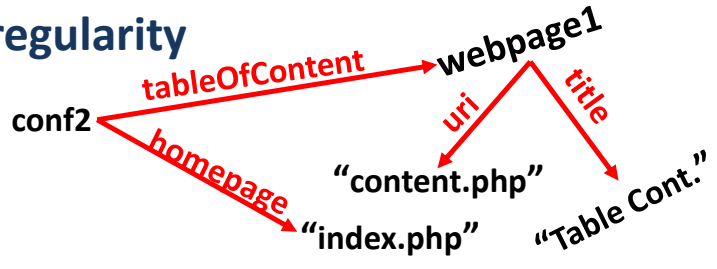



| ID | type | creator | title | partOf |
|---|---|---|---|---|
| inproc1 | inproceeding | {author3, author4} | "AAA" | conf1 |
| inproc2 | inproceeding | author2 | "BBB" | conf1 |
| inproc3 | inproceeding | author3 | "CCC" | conf2 |

Foreign Key Relationship

| ID | type | title | issued |
|---|---|---|---|
| conf1 | Conference | "conference1" | 2010 |
| conf2 | Proceedings | "conference2" | 2011 |

Irregularity




Example of structure recognized from RDF graph

# What does “schema” mean?

## Relational Schema

Describes the structure of the occurring data

Concept mixing (for convenience)

Designed for one database (=dataset)

## Semantic Web Schema

Purpose: knowledge representation

Describing a concept universe (regardless data)

Designed for interoperability in many contexts

Statement: it is useful to have **both** an (Emergent) Relational and Semantic Schema for RDF data

- useful for systems (higher efficiency)
- useful for humans (easier query formulation)

# When is a Emergent Schema of RDF data useful?

- **Compact** Schema
  - as few tables as possible
  - homogeneous literal types (few NULLs in the tables)
- **Human-friendly** "Labels"
  - URIs + human-understandable table/column/relationship names
- High "**Coverage**"
  - the schema should match almost all triples in the dataset
- **Efficient** to compute
  - as fast as data import

CWI

Basic CS discovery

*(s1, offers, offer1)*
*(s1, region, region1)*
*(s2, offers, offer2)*
*(s2, offers, offer3)*
*(s2, region, region1)*
*...*
*(offer1, availableDeliveryMethods, DHL)*
*(offer1, description, "Offer data")*
*(offer1, hasBusinessFunction, "Sell")*
*(offer1, hasEligibleQuantity, 1)*
*(offer1, hasInventoryLevel, 1)*
*(offer1, hasStockKeepingUnit, 112)*
*(offer2, availableDeliveryMethods, DHL)*
*(offer2, hasPriceSpec, price1)*
*(offer2, hasStockKeepingUnit, 112)*
*(offer2, type, Offering)*
*...*
*(price1, hasCurrency, "EUR")*
*(price1, hasCurrencyValue, "35.99")*
*(price1, hasUnitOfMeasurement, "C62")*
*(price1, valueAddedTaxIncluded, "false")*
*(price1, eligibleTransactionVolume, 0)*
*(price1, ...*
*...*

**<Example RDF triples>**

CWI

# Characteristic Sets in some well-known RDF datasets

| Datasets | #triples* | #CS's | #CS's to cover 90% | Avg. #prop. | #multi-type properties |
|---|---|---|---|---|---|
| LUBM | 100M | 17 | 7 | 5.71 | 0 |
| BSBM | 100M | 49 | 14 | 12.61 | 0 |
| SP2Bench | 100M | 554 | 7 | 9.8 | 0 |
| **synthetic** | *data created by benchmark data generator* | | | | |
| MusicBrainz | 179M | 27 | 10 | 4.7 | 0 |
| EuroStat | 70K | 44 | 8 | 7.77 | 0 |
| DBLP | 56M | 249 | 8 | 13.70 | 0 |
| PubMed | 1.82B | 3340 | 35 | 19.27 | 0 |
| **relational** | *RDF data from a relational database dump* | | | | |
| WebData. | 90M | 13354 | 930 | 7.94 | 551 |
| DBpedia | 404M | 439629 | 85922 | 24.36 | 1507 |
| **native** | *real data originating as RDF* | | | | |

# Partial and Mixed Use of Ontologies

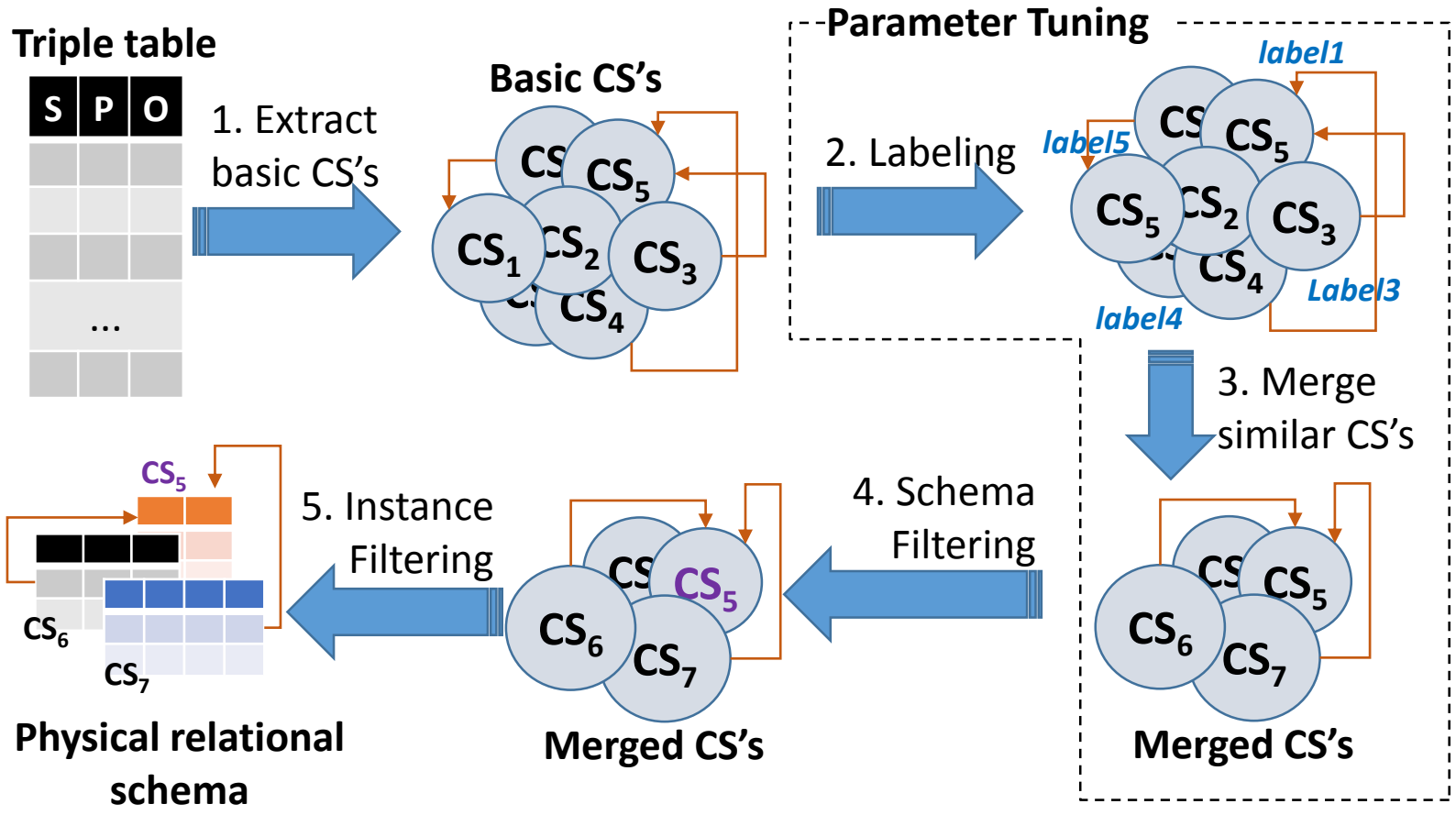
| | mixed | |
|---|---|---|
| dataset | number of ontology classes used per CS | % I us |
| LUBM | 1.94 | |
| BSBM | 3.96 | |
| SP2Bench | 4.94 | |
| MusicBrainz | 3.93 | |
| EuroStat | 3.14 | |
| DBLP | 6.58 | |
| PubMed | 4.94 | |
| WebData. | 2.27 | |
| DBpedia | 8.35 | |

| $cs_4$ |
|---|
| dc:description |
| gor:validFrom |
| gor:validThrough |
| gor:hasCurrency |
| gor:hasCurrencyValue |
| gor:hasUnitOfMeasurement |
| gor:valueAddedTaxIncluded |
| gor:eligibleTransactionVolume |

*(prefix gor: http://purl.org/goodrelations/v1#*
*prefix dc: http://purl.org/dc/elements/1.1/)*

| PriceSpecification |
|---|
| gor:description |
| gor:name |
| gor:eligibleTransactionVolume |
| gor:validFrom |
| gor:validThrough |
| gor:hasCurrency |
| gor:hasCurrencyValue |
| gor:hasUnitOfMeasurement |
| gor:valueAddedTaxIncluded |
| gor:hasMaxCurrencyValue |
| gor:hasMinCurrencyValue |

# Emerging a Relational Schema



CWI

# Results: compact schemas with high coverage

| | Number of tables | | | Coverage – Metric $C$ (%) | | |
|---|---|---|---|---|---|---|
| Datasets | before merging | after merging | remove small tables | remove small tables | prune infreq. prop. | final schema |
| LUBM | 17 | 13 | 12 | 100 | 100 | **100.00** |
| BSBM | 49 | 8 | 8 | 100 | 100 | **100.00** |
| SP2B | 554 | 13 | 10 | 99.99 | 99.65 | **99.65** |
| MusicBrainz | 27 | 12 | 12 | 100 | 99.9 | **99.60** |
| EuroStat | 44 | 10 | 5 | 99.73 | 99.53 | **99.53** |
| DBLP | 249 | 9 | 6 | 100 | 99.68 | **99.60** |
| PubMed | 3340 | 14 | 12 | 100 | 99.75 | **99.73** |
| WebData. | 13354 | 3000 | 253 | 98.17 | 94.37 | **92.79** |
| DBpedia | 439629 | 542 | 234 | 99.12 | 96.68 | **95.82** |

# Results: understandable labels & performance

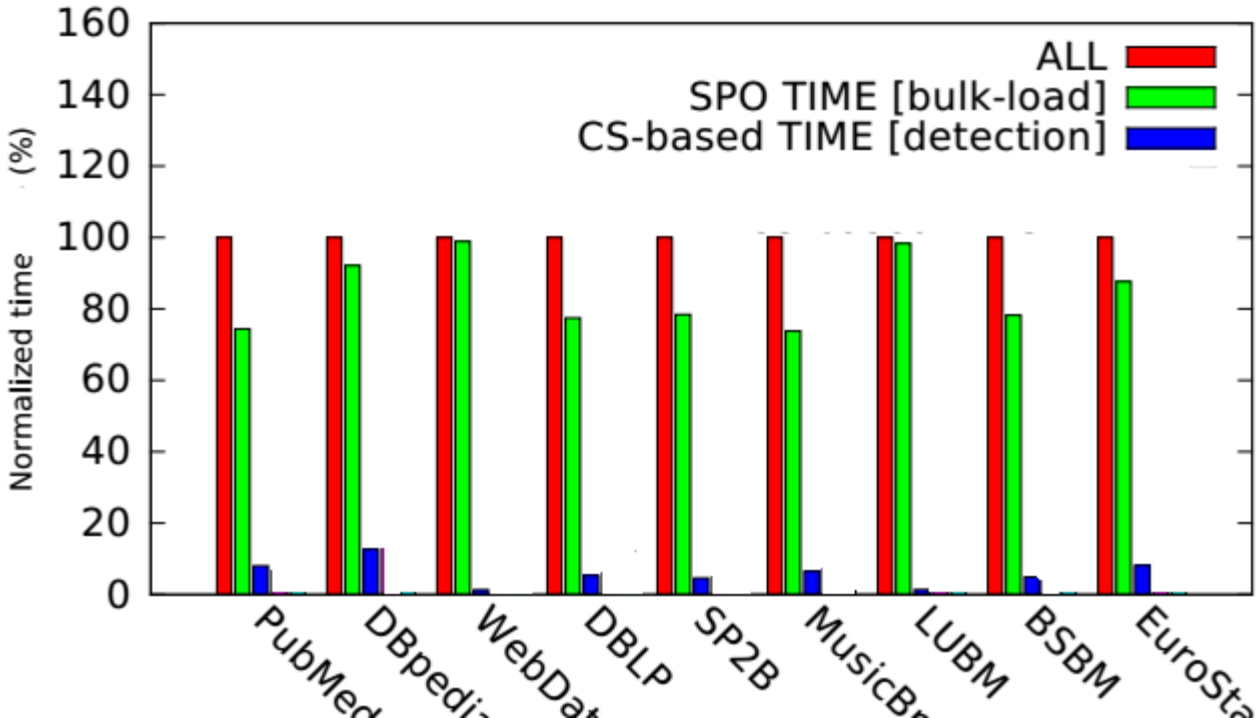
| labels | WebData. | DBpedia |
|---|---|---|
| top 3 | 3.6 | 3.8 |
| final | 4.1 | 4.6 |

Table 3: Human survey results on Likert scale

Likert Score: 1=bad ….. 5=excellent

| RDF Store | Query 3 | | | Query 5 | | |
|---|---|---|---|---|---|---|
| | Cold | Hot | Opt. Time | Cold | Hot | Opt. Time |
| Virt-Quad | 4210 | 53 | 40.2 | 3842 | 1350 | 18.6 |
| Virt-CS | 2965 | 9 | 5.4 | 2130 | 712 | 4.2 |

Table 5: Query time (msecs) w/wo the recognized schema
(Cold: First query runtime after re-starting the server
Hot : Run the query 3 times and get the last runtime
Opt. Time: Query optimization time)




Thank
You!