Towards Linked Data Update Notifications
Reviewing and Generalizing the sparqlPuSH approach
NoISE Workshop at ESWC 2015, June 01, 2015

Magnus Knuth¹, Dinesh Reddy¹, Anastasia Dimou², Sahar Vahdati³, and George Kastrinakis⁴
¹ Hasso Plattner Institute – University of Potsdam, Potsdam, Germany
² Ghent University - iMinds, Ghent, Belgium
³ Institute of Computer Science III – University of Bonn, Bonn, Germany
⁴ National Technical University of Athens, Athens, Greece
Agenda

- Motivation
  - Notification – the sparqlPuSH idea
  - Shortcomings – the sparqlPuSH implementation
- Requirements analysis
- Open Research Problems
- Conclusion
How can data consumers be supported when working with changing Linked Data datasets?

Motivation
Notification - The sparqlPuSH idea [Passant2010]

- consumer/subscriber gets notified on SPARQL query result change
  - subscribers register their SPARQL queries
  - on result change notify subscribers
    - send Push notification via PubSubHubbub
    - tracking result changes within RSS/Atom feed
Shortcomings – The sparqlPuSH implementation [Passant2010]

- intended to be used for micro-blogging notifications

- restrictions in SPARQL queries
  - only SELECT
  - ?uri and ?date variables mandatory
  - ?label and ?author variables optional
- max ?date binding serves as the change indicator
- dataset updates only via the interface
Requirements

■ (R1) Processing arbitrary SPARQL queries
  □ individual queries can be formulated by subscribers

■ (R2) Application on any accessible SPARQL interface
  □ any queryable dataset can be used

■ (R3) Avoidance of unnecessary load to SPARQL interfaces
  □ querying is expensive, so don’t DoS public endpoints

■ (R4) Sufficiently expressive description of what has changed
  □ the feed of changes should contain detailed information
Open Research Problem (P1) – Handling large SPARQL query results (R1, R2, R3)

How can the SPARQL result size be limited efficiently without losing relevant information for change detection?

- **Aggregation**, e.g. rewrite SELECT queries as
  ```sql
  SELECT COUNT(*) WHERE ...
  ```
  - way smaller result size, but incomplete

- **Hashing**, store only hash value of result instead of result itself
  - reduced storage, still full result to transfer

- **Streaming** as applied by Triple Fragments servers
  - detect changes without retrieving the full result
How can the SPARQL results be effectively and efficiently compared?

- **ASK** queries deliver a boolean
- **DESCRIBE** and **CONSTRUCT** queries’ result is an RDF graph
  - graph isomorphism problem, ongoing research on canonical labeling of blank nodes [Hogan2015]
- **SELECT** queries deliver result set (variables + solutions/bindings)
  - Jena ARQ provides ResultSetCompare (without diff)
  - internally result sets are transformed to RDF (see above)
Open Research Problem (P3) - Scheduling the re-evaluation of SPARQL queries (R3)

How to determine the best time at which a relevant data change has occurred?

- **Dataset descriptions**, e.g. using VoID or DCAT vocabulary
- **Sensoring updates** from publishers
  - still not trivial to compute the necessity of re-evaluation of particular SPARQL queries based on full changesets
- **Estimating update intervals**
  - e.g. based on change history
Open Research Problem (P4) - Equality of SPARQL queries (R3)

*Which of the following SPARQL queries are similar equivalent? Why?*

[Dividino2013]

- **Syntactical Query Similarity**
  - transform queries to canonical form
  - Levenshtein similarity is not enough

- **Structural Query Similarity**
  - transform queries to pattern trees and check for equivalence or containment
Open Research Problem (P5) - Describing changes in SPARQL query results (R4)

How can changes of SPARQL query results be described in an extent useful to the end user?

- **Describe transition between to result sets**, e.g. as RDF change
- multiple vocabularies available: Changesets, Graph Update Ontology, RDF Patch
Conclusion

- **Requirements**
- **Issues**
  - non-trivial research problems

Regarding performance and scalability there might not be an ultimate solution
- solutions depend on the given setting
  - dataset characteristics
  - dataset update mechanisms
  - individual notification requirements
Any Questions?

Magnus Knuth
Hasso Plattner Institute, University of Potsdam, Germany
mail  magnus.knuth@hpi.de
tel    +49 331 5509 547
webID http://magnus.13mm.de/