List.MID: A MIDI-Based Benchmark for RDF Lists

Albert Meroño-Peñuela, Vrije Universiteit Amsterdam
@albertmeronyo
Enrico Daga, The Open University

ISWC 2019, 28 October, Auckland
Modelling choices are often left to **subjective choice**

These practices and their reuse are key in **query performance**

**Lists** are everywhere! Co-authors, timelines, media, recipes, etc.
And in **MIDI** applications

...  
[
144, 60, 100
]
[
128, 60, 64
]
...

- So what do we know about performance of **RDF List** solutions?
 RELATED WORK

- Modelling of RDF lists
  - RDF(S) container classes (rdf:Bag, rdf:Alt, rdf:Seq)
  - Closed collections (rdf:List, rdf:first, rdf:rest, rdf:nil)
  - JSON-LD/Turtle syntaxes: "@list": [ "joe", "bob", "jaybee" ],
    :a :b ( "bob" "alice" "carol"
  - Ontology Design Patterns: Sequence OP, Collections Ontology

- Benchmark datasets and queries
  - BSBM, LUBM, SP²Bench, DBPedia SPARQL, WatDiv
  - LSQ
  - IGUANA, LDBC
RESEARCH QUESTIONS & CONTRIBUTIONS

What RDF list models are common in LOD? How can we generate data and queries to test their impact at retrieval in a systematic and principled manner?

C1: Survey of common list modelling practices in RDF and LOD
C2: Benchmarking data and queries enabling their comparison
C3: Evidence of present and future use
Surveyed from:

- W3C standards
- The Ontology Design Patterns portal
- List choices in RDF datasets from ISWC resource track papers
- Linked Open Vocabularies (LOV)
- LOD Laundromat/LOD-a-lot [Fernández et al. ISWC 2017]

Findings: various practices that generalize to 6 list modelling patterns
RDF SEQUENCE AND RDF LIST

[SEQ]

[LIST]
SEQUENCE ONTOLOGY PATTERN

[SOP]

:e₁

sequence:precedes

sequence:follows

:e₂

sequence:precedes

sequence:follows

:e₃
LIST.MID is an RDF list benchmark based on:

- **Collaborative GitHub awesome MIDI list**
- More than **300K MIDI files** from the Web
- **MIDI Linked Data Cloud** [Meroño et al. ISWC 2017]

Consists of:

- **RDF List data generator** on top of real-world list data
- Collection of extensible **SPARQL list-retrieval queries**
Custom version of **midi2rdf algorithm** [Meroño et al. ESWC 2016]

```bash
midi2rdf [-h]
  [--format [{xml,n3,turtle,nt,pretty-xml,trix,trig,nquads,js
```
CQ1. Full list lookup: What is the ordered content of the list?
CQ2. N-th Lookup: Which is the n-th item in the list?
CQ3. Ordered Range: What are the n...m items in the list?

Aimed at supporting use-case LOD publishing
Focus on minimal and atomic operations related to list ordered access
Do not deal (yet) with list management (edit, merge, split, etc.)
LIST.MID QUERIES

BIND (xsd:integer(SUBSTR(str(?seq), 45)) AS ?index) 
} ORDER BY ?index

[LIST] SELECT ?event (COUNT(?step) as ?index) WHERE {
 :list a midi:Track ; midi:hasEvents ?events . ?events rdf:rest* ?step .
} GROUP BY ?event ORDER BY ?index

[URI] WHERE { [] a midi:Track ; midi:hasEvent ?event .
BIND (xsd:integer(SUBSTR(str(?event), 77)) AS ?id) } ORDER BY ?id

[NUM/TIME] WHERE { [] a midi:Track ; midi:hasEvent ?event .
?event midi:absoluteTick ?tick . } ORDER BY ?tick

BIND (xsd:integer(SUBSTR(str(?event), 77)) AS ?id) 
} ORDER BY ?id
EVALUATION: PERFORMANCE EXPERIMENTS

[Daga et al. QuWeDa 2019]

- rdf:List has poor performance
- rdf:Seq trade-off

Blazegraph

- Property-based better than link-based

Coherent results among triplestores
Critical impact of different models
EVALUATION: PERFORMANCE EXPERIMENTS

N=24 in W3C semantic-web, public-lod, VU, OU
CONCLUSIONS

Lists are important! But how to assess the impact of their models?

- 6 common **list patterns** in RDF
- **List.MID**: Real-world, MIDI-based RDF list benchmark **data generator**
- **List.MID**: SPARQL queries for minimal and atomic list **retrieval operations**
- First **experiments** and community interest

Limitations/future work:

- Limited set of list operations (e.g. `rdf:List` could win in e.g. addition)
- Extensions to other list models
- Recommendations for triplestore optimization implementations
Questions, comments, suggestions most welcome

@albertmeronyo
@enridaga
https://github.com/MIDI-LD/List.MID
• Motivation
• Related Work
• List Pattern Survey
• List.MID Benchmark Data Generator
• List.MID SPARQL Queries
• Evaluation: Performance Experiments
• Evaluation: Survey of Interest
• Conclusions
Use this slide to place an image to the left and text to the right.

- With bullet
  - Secondary list

To replace the image, right-click the image (click on it with your other mouse button), select Change picture... and choose the new image.