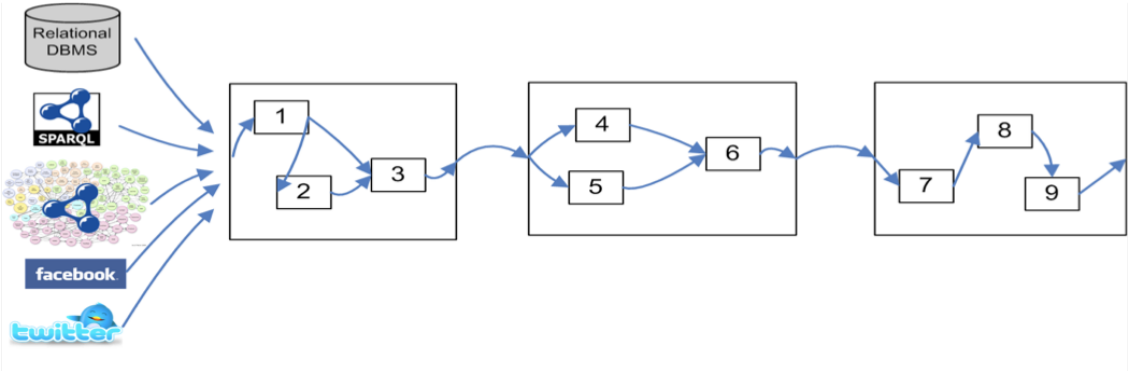
The RDFGears Project

Jan Hidders

WISer Day 2012

June 12, 2012

Goal of RDFGears Project



Goal: building a language and system for RDF transformations

- XML: XSLT & XQuery, RDF: ?? & SPARQL
- Visual workflows, easy to use, extensible
  - Inspired by scientific workflow systems: Taverna, Kepler, Triana, Galaxy, ...
- Well-founded and well-understood language allowing research into:
  - Scalability, optimization, provenance, ...
- Applications: (RDF) Data
  - transformation
  - integration and enrichment
  - publication

Workflow Languages for RDF transformation

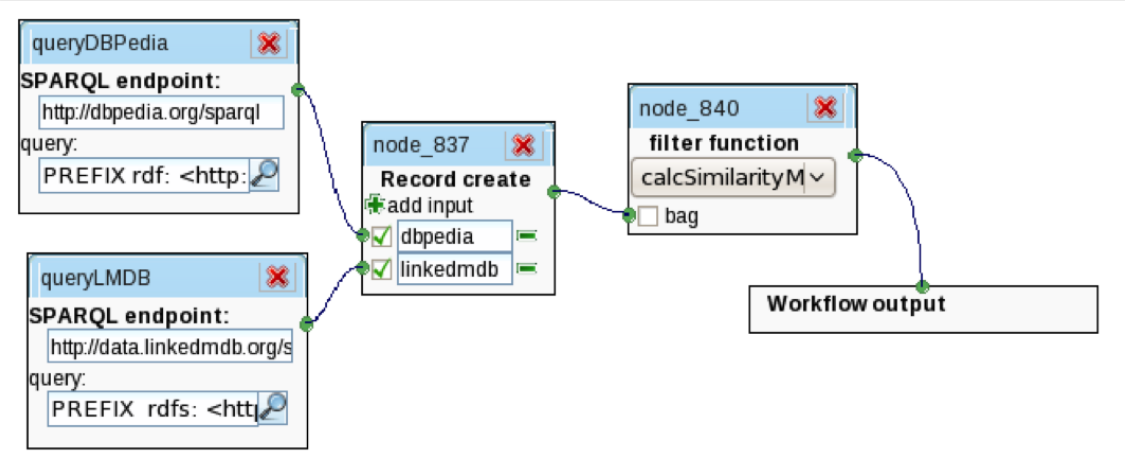
What is a suitable (graphical) workflow language?

- Expressive power, as graph transformation language
- Understandability and Usability
- Versatility and Extendability
- Optimizability

Our solution:

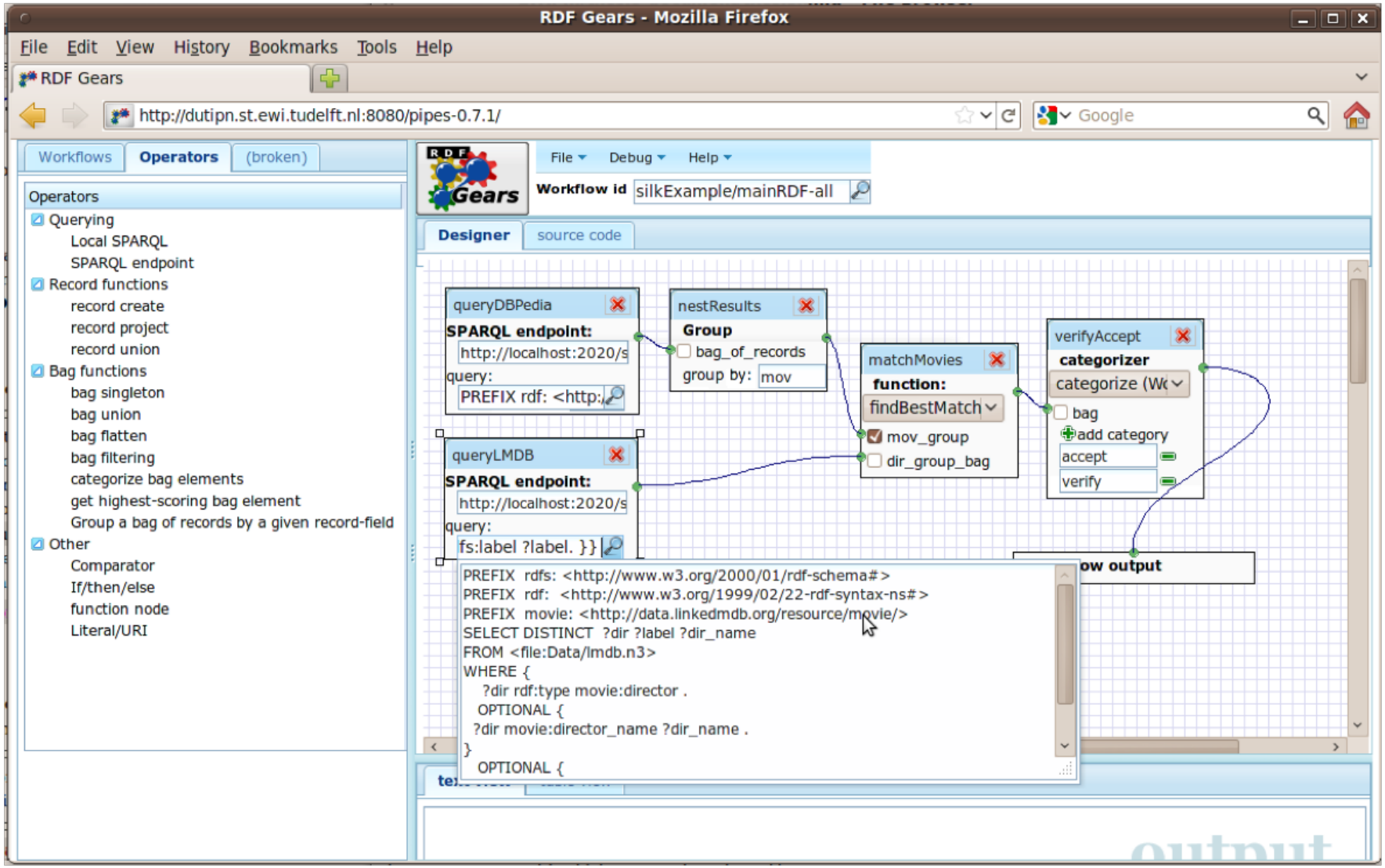
- Start from Nested Relational Algebra with a graphical notation
- Add basic types: Literal, URL, RDF-Graph
- Use SPARQL for (1) retrieval and (2) graph transformation/creation

The RDFGears Workflow Language



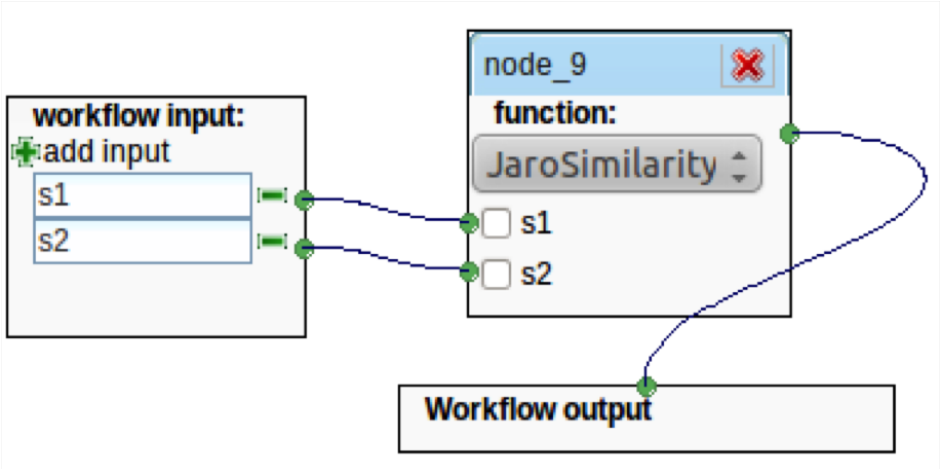
- A **SPARQL endpoint** processor produces a bag of tuples.
  - Missing variables in binding are represented by null values.
- A **Record create** processor creates a record.
  - Input ports can be marked as "iterating" which means it consumes a bag rather then a single value and iterates over that bag
  - If multiple ports are marked, cartesian product is computed.
- A **Filter** processor selects elements from a bag based on a user-defined function or earlier created workflow.

Nested Relational Calculus/Algebra operators



- Usual operators from NRC/NRA available, plus short-hands.

Workflows published as web services



- All user-defined workflows can be published as web service.
- `http://wis.ewi.tudelft.nl/rdfgears/myworkflow?s1=Good&s2=Goed`
- Output is RDF graph or xml-encoded nested relational value.

Ongoing and Future Research Topics

- Streaming main-memory implementation.
- Typing.
- Disk-based implementation.
- Query optmization by graph rewriting
- Distributed query optimization
  - Leverage NRA/NRC rewriting rules
  - Eliminate and optimizate SPARQL queries
  - Allow virtual SPARQL endpoint publishing of workflow results
- MapReduce mapping of RDFGears Workflow Language
  - In general interesting question for NRC/NRA
  - What is a good mapping? What is a complete mapping?
  - Also related to already existing similar languages: Pig, Hive
- Provenance functionality
  - Differences between NRA and NRC?
  - Why provenance. Why-not provenance.