Semantic Technologies for Data Analysis in Health Care

Robert Piro† Peter Hendler‡ Ian Horrocks†

Motivation

US Health Care Organisations (HCOs) are compared by measuring Quality of Care:

- A quality measure is a percentage of a selected population

$$\frac{\#\text{diabetic patients with eye exams}}{\#\text{diabetic patients}} \times 100\%$$

- NCQA (National Committee of Quality Assurance) maintains precise specs for quality measures. E.g. HEDIS.
- HCOs must annually publish HEDIS measurements
- HEDIS is used to accredit HCOs for billing against government funded health care schemes which cover approx. 20% of the US population.

Computing HEDIS

- HEDIS is very complex
- Quality measures require complex analysis of the data
- Heterogeneous data sources

Current State of Affairs

Solutions used to compute HEDIS

- Vendor solution
  - Acts as "Black Box"
  - Results difficult to validate
- In-house solution: combination of SAS programs and SQL queries
  - Complex and inefficient
  - Difficult to maintain
  - Development is a drain on resources

Semantic Technology Approach

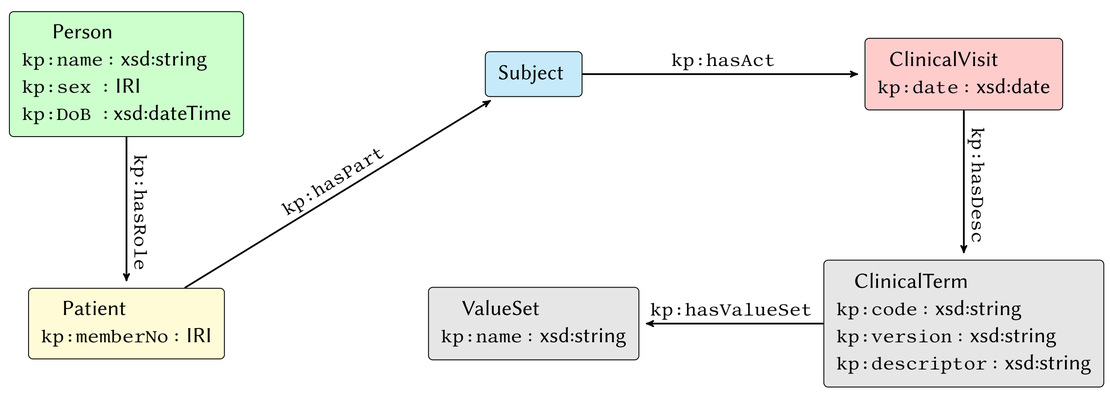
The Data Model

- RDF Data Format
  - ▷ Easily extensible and flexible
  - ▷ Ideal to integrate data from the heterogeneous data sources
- Schema ontology designed according to HC Informatics Standards (HL7 RIM)
  - ▷ Close to domain expert conceptualisation
  - ▷ Familiar to domain experts

Encoding HEDIS measures

- Encoding in Datalog rules.
  - ▷ Intuitive if-then-statements — legible
  - ▷ Purely declarative — no procedural statements
  - ▷ Succinct — 174 rules vs 3000 lines SQL
  - ▷ General recursion — not avail. in SQL

Schema Ontology derived from HL7 RIM (Excerpt)



RDF-Triples of a data instance:

PREFIX kp: <http://www.kp.org/>

kp:Person/4711 kp:name "John Smith"
kp:Person/4711 kp:sex kp:Male
kp:Person/4711 kp:DoB "1983-07-15"
kp:Person/4711 kp:hasRole kp:Patient/4711
kp:Patient/4711 kp:hasRole kp:Patient/4711
kp:Patient/4711 kp:hasPart kp:Subject/4711
kp:Subject/4711 kp:hasAct kp:CV/1503
kp:CV/1503 kp:hasDate "2015-10-31"
kp:CV/1503 kp:hasDesc kp:CT/250.70
kp:CT/250.70 kp:code "250.70"
kp:CT/250.70 kp:version "ICD9"
kp:CT/250.70 kp:descriptor "Diabetes with..."
kp:CT/250.70 kp:hasVS kp:VS/DiabDiag
kp:VS/DiabDiag kp:name "Diabetes Diagnosis"

Encoding HEDIS CDC

HEDIS Comprehensive Diabetic Care (CDC) is the most complex chapter of HEDIS

Quote from HEDIS

'[Diabetics are those patients] who met any of the following criteria during the measurement year [2013] or the year prior to the measurement year [2012] (count services that occur over both years):
At least two outpatient visits (Outpatient Value Set), observation visits (Observation Value Set) or nonacute inpatient visits (Nonacute Inpatient Value Set) on different dates of service, with a diagnosis of diabetes (Diabetes Value Set). Visit types need not be the same for the two visits."

RDFox-Datalog Rules (partly encoding the quote)

```
[?CV,rdf:type,aux:diabetesDiagnosis]  :-
      [?CV,kp:hasDesc,?CT], [?CT,kp:hasValueSet,?VS],
      [?VS,kp:name,"Diabetes Diagnosis"] .

[?pat,aux:admissibleVisit,?CV]  :-
      [?pat,aux:patientHasAct,?CV],
      [?CV,rdf:type,aux:outpatient],
      [?CV,rdf:type,aux:diabetesDiagnosis] .
```

Similarly one defines rules for `aux:outpatient`, `aux:nonacute-inpat`, etc.

Challenges with HEDIS CDC

HEDIS CDC requires

- Recursion: "Eligible are only those patients with *continuous enrollment*"
- Negation: "Exclude all members that have ..."
- Aggregation (max/min): "Report the latest and 'best' blood pressure measurement.
- Non-tree shaped rules: (HEDIS quote above)
- Value manipulations: date ⇝ year

Benefits using RDFox-Datalog

- Rules can be authored by non-IT experts
  - ▷ Tools for authoring and data-browsing
  - ▷ Easy creation of "views"
  - ▷ Efficient execution automated by RDFox
  - ▷ Explanations/Justifications by RDFox: Results can be traced through the rules to the raw data (proof tree)
- Powerful Expressivity
  - ▷ FILTER and BIND constructs
  - ▷ Stratified Negation as Failure (NAF)
  - ▷ General recursion
  - ▷ Enough for HEDIS CDC!

RDFox

RDFox is an in-memory RDF-triple store and parallel Datalog reasoner (Linux/OSX/Win)

- C++/Java/Python API
- low memory per triple
- explanation function
- SPARQL endpoint
- high scalability
- stratified NAF

`www.rdfox.org`

Evaluation

- Commodity Hardware: Linux Server 8 Intel Xeon @2.7GHz and 64GB RAM
- 10GB patient data in 100 Million records
- Translation with a Scala (Java) application:
  - ▷ run time: 45min on 8 cores
  - ▷ resulting RDF-graph: 293M triples
- Data Import with RDFox: 11min on 8 cores using 18GB (28% RAM)
- Computation of HEDIS CDC measures:
  - ▷ RDFox run-time: 19min on 8 cores

Conlusion

- Standards-based, unified data model
  - ▷ Shortened development time in rule authoring phase
  - ▷ Higher acceptance by domain experts
- RDFox-Datalog rules
  - ▷ Intuitive — less modelling errors
  - ▷ Legible & succinct — better maintainable
- RDFox
  - ▷ Computes in competitive time
  - ▷ Explanations — reduce development cycles

Acknowledgements

This project was jointly funded by the DBOnto Platform Grant (EPSRC, EP/L012138/1) and Kaiser Permanente. Funds from Kaiser Permanente were contributed by Patrick Courneya and Andy Amster. Many thanks go to Alan Abilla at Kaiser Permanente's CMT Big Data Modelling Project to second Peter Hendler to the project. Paul Glenn at Kaiser Permanente Georgia kindly seconded Scott Kimberly to this project and approved the use of the sensitive data without which the project had been pointless. Thanks also go to Joseph Jentzsch for bringing the funding together as well as Mike Sutten and his group who funded the research stay at the Kaiser Permantente Technology Campus in Pleasanton and organised and provided the technical support for this project.

peter.hendler@gmail.com‡ forename.lastname@cs.ox.ac.uk†