Semantic Technologies for Data Analysis in Health Care

Robert Piro Ian Horrocks

University of Oxford

Oslo, May 2016

Overview

1 The Project

2 Motivation

3 Solution

4 Encoding Data in RDF

5 Encoding of HEDIS CDC

6 Evaluation

Project jointly funded by DBonto and Kaiser Permanente

DBONTO

- EPSRC funded “platform” at University of Oxford
- Funds exploratory projects with industry collaborators

KAISER PERMANENTE

- US “Health Maintenance Organisation” (HMO)
- Largest ‘managed care’ organisation in the US with 10.2M members
- Active in 8 US regions with 195 000 employees
- Turn over 56.4bn US$ and net income 3.1bn US$

Quality Measures in US Health Care

- HMOs are obliged to deliver information on their quality of care
- The National Committee of Quality Assurance (NCQA) maintains specifications for quality measures, e.g., HEDIS.
- A quality measure is a percentage of a selected population, e.g.:

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

- HEDIS is used to accredit HMOs for billing against government funded health care schemes which cover approx. 20% of the US population.

$\Longrightarrow$ HMOs have a big incentive to deliver prompt and accurate data

Challenges with HEDIS

- HEDIS is a very complex specification (examples later)
- Quality measures require complex analysis of the data
- Data needs to be assembled from heterogeneous data sources

Current state of affairs

- Either a combination of SAS programs and SQL queries (in-house) or a vendor product is used
- Solutions are complex, inefficient and difficult to validate and maintain.
- Several ad-hoc invented schemas are used independently

Semantic Technology Approach

The Data Model

- RDF is used to integrate data from the heterogeneous data sources
- An ontology is used to describe a flexible but largely uniform schema
- Schema ontology designed according HL7 RIM standard (familiar to domain experts)

Encoding HEDIS measures

- Datalog rules are used to encode the HEDIS specification
- RDFox triple-store/Datalog-engine used to compute consequences of RDF-data+rules
- Simple SPARQL counting queries used to finally compute quality measures

Semantic Technology Approach

Evaluation

- Encoded the most complex subsection of the HEDIS measures: Comprehensive Diabetic Care (CDC).
- Translated the patient history of KP Georgia region (466 000 patients) into RDF-triples
- Loaded RDF-triples into RDFox, materialised Datalog rules and ran SPARQL queries
- Compared results with those produced by existing vendor solution
- Used RDFox explanation capability to investigate differences

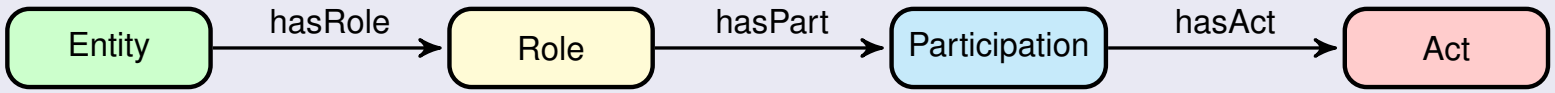
Design Goals

The model should be:

- Close to the domain expert conceptualisation of the described data
- Flexible and uniformily capture the health care data
- Amenable to Semantic Technologies, i.e. encodable in RDF

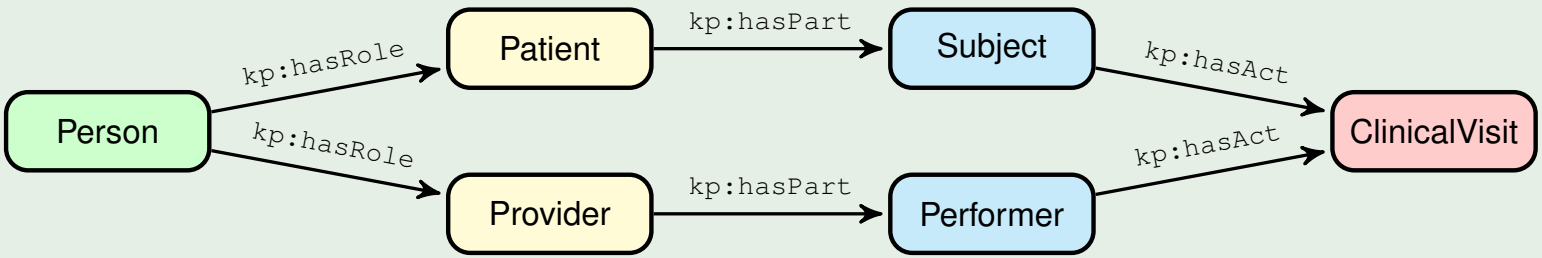
ERPA paradigm

- Describes business processes as 'Enities in Roles Participating in Acts'



- Derived from HL7 RIM standard for modelling data in healthcare informatics

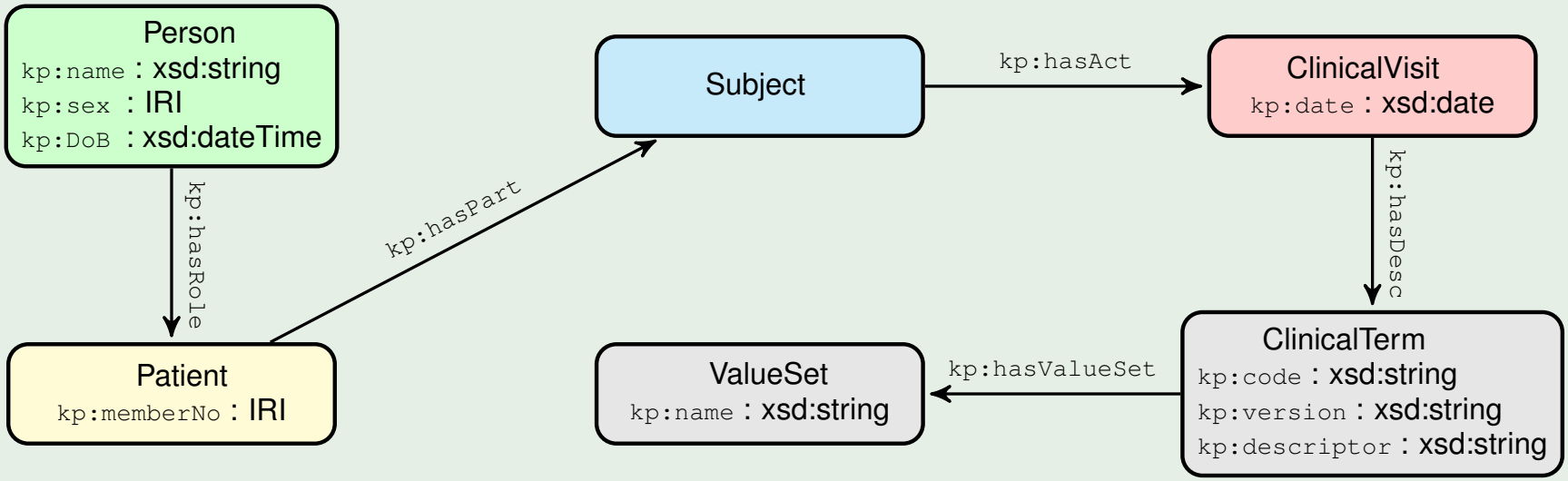
Example (Clinical Visit)



Clinical visit

- Graph shows typical ERPA colour scheme: green (Entity), yellow (Role), blue (Participation), red (Act)
- E.g. Patient and Provider can be thought as OWL-subclass of Role
- `kp:hasRole` corresponds to an RDF/OWL property

Example (Expansion of the 'Patient-branch' in Clinical Visit)



RDF-triples for clinical visit with ICD9 diagnosis code 250.70

```
<http://www.kp.org/Patient/memberNo> kp:hasPart <http://www.kp.org/Subject/UID1> .
<http://www.kp.org/Subject/UID1> kp:hasAct <http://www.kp.org/Visit/UID2> .
<http://www.kp.org/Visit/UID2> kp:date "2013-09-10T00:00:00"^^xsd:dateTime .
<http://www.kp.org/Visit/UID2> kp:hasDesc <http://www.kp.org/CT/250.70> .
```

ClinicalTerms and ValueSets

RDF-triples for the ClinicalTerm 250.70

```
<http://www.kp.org/CT/250.70> kp:code "250.70" .
<http://www.kp.org/CT/250.70> kp:version <http://www.kp.org/Version/ICD9> .
<http://www.kp.org/CT/250.70> kp:descriptor "Diabetes with peripheral circula..." .
<http://www.kp.org/CT/250.70> kp:hasValueSet <http://www.kp.org/ValueSet/DD> .
```

RDF-triples for ValueSet "Diabetes"

A Value Set is a set of codes (ICD9/ICD10/...)

- The code range 250.00 - 250.99 encodes different types of diabetes
- The value set "Diabetes" contains all codes from 250.00 to 250.99

```
<http://www.kp.org/ValueSet/DD> kp:name "Diabetes" .
<http://www.kp.org/CT/250.00> kp:hasValueSet <http://www.kp.org/ValueSet/DD> .
...
<http://www.kp.org/CT/250.99> kp:hasValueSet <http://www.kp.org/ValueSet/DD> .
```

Advantages

Advantages of using RDF/OWL schema

- Single coherent schema for all the healthcare data involved
- RDF schema is easily extensible
- Based on modelling standards developed in healthcare informatics
- Familiar to domain experts

Example (Quote from the Definition of a Diabetic Patient)

[*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.*
- *. . .*

Assembling the necessary information using rules

```
[?CV, rdf:type, aux:outpatient] :-[?CV, kp:hasDesc, ?PT],
                                  [?PT, kp:hasValueSet, ?VS],[?VS, kp:name, "Outpatient"] .

[?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] .
```

Example (Quote from the Definition of a Diabetic Patient)

[*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.*
- *. . .*

Encoding "diabetic patient" rule

```
[?pat, rdf:type, aux:diabeticPatient]:-
             [?pat, aux:admissibleVisit, ?CV0],    [?pat, aux:admissibleVisit, ?CV1],
             [?CV0, kp:date, ?date0],              [?CV1, kp:date, ?date1],
             BIND( YEAR(?date0) AS ?y0 ),          BIND( YEAR(?date1) AS ?y1 ),
             [kp:HEDIS, kp:measurementYear, ?y0], [kp:HEDIS, kp:measurementYear, ?y1]
             FILTER ( ?date0 != ?date1 ).
```

- Rule is non-treeshaped and thus not expressible in OWL RL
- Uses value manipulations (BIND constructs)
- Uses date comparisons (FILTER constructs)

Example (Exclusions of Patients)

Exclude members [from the pop. of interest] who meet any of the following criteria:

- *IVD [Ischemic Vascular Disease]. Members who met at least one of the following criteria during both the measurement year and the year prior to the measurement year. Criteria need not be the same across both years.*
  - *At least one outpatient visit (Outpatient Value Set) with an IVD diagnosis (IVD Value Set).*
  - *At least one acute inpatient encounter (Acute Inpatient Value Set) with an IVD diagnosis (IVD Value Set).*
- *. . .*

Calculating Negated Properties

- Requires some kind of negation — in fact negation as failure (NAF)
- As a work-around:
  - We compute all patients with IVD according to the specification
  - We used a SPARQL-query with FILTER NOT EXISTS construct to compute non-IVD patients
  - We fed this information back into RDFox and continued the computation

Continuous Enrolment

- Patients eligible for measurment must be enrolled with HMO and can have multiple enrollements in a year.
- Enrollments are given as [begin-date,end-date] pair per patient.
- The enrollments must form a connected chain

$$[x_0, x_1] \ldots [x_i, x_{i+1}][x_{i+1}, x_{i+2}] \ldots [x_{n-1}, x_n]$$

such that “2013-01-01” and “2013-12-31” are enclosed by some interval

Example (Recursion)

“Compute all patients with contintuous enrollments within the measurement year”

```
[?pat, aux:contEnrollment, ?Enr]:-
          [?pat, kp:hasEnrollment, ?Enr], [?Enr, kp:beginDate, ?bDate],
          [?Enr, kp:endDate, ?eDate],     FILTER(?bDate <= "2013-01-01" <= ?eDate ) .
[?pat, aux:contEnrollment, ?Enr]:-
                [?pat, aux:contEnrollment, ?PredEnr], [?PredEnr, kp:endDate, ?date],
                [?pat, kp:hasEnrollment, ?Enr],       [?Enr, kp:beginDate, ?date] .
```

Aggregation

- Aggregate functions compute from multiple values a single value like *max*.
- Many HEDIS CDC measures ask for the best (min/max) and latest value (date maximal)

Example (Quote from Blood Pressure (BP) Measurement)

[. . .] identify the **most recent** *BP reading taken during an outpatient visit (Outpatient Value Set) or a non-acute inpatient encounter (Nonacute Inpatient Value Set) during the measurement year. The member is numerator compliant if the BP is <140/80 mmHg. [. . .] If there are multiple BPs on the same date of service, use the* **lowest** *systolic and* **lowest** *diastolic BP on that date as the representative BP.*

Computation of the latest

- We marked BP readings of a patient if there was a reading at a later date.
- We used a SPARQL query with FILTER NOT EXISTS to determine the reading that had *no* mark — the latest reading.
- The results were fed back to RDFox and the computation continued

Advantages of Using RDFox-Datalog Rules

- Intuitive if-then-statements are (relatively) easily legible
- Purely declarative — no procedural statemtents
- Succinct 174 Rules instead of $>$3000 lines of SQL code
- RDFox-Datalog rules provide
  - Recursion. Not possible with SPARQL/SQL queries alone
  - BIND and FILTER constructs — which are needed in our encoding

RDFox Extensions since the Project

- RDFox-Datalog has been extended with (stratified) NAF and (stratified) aggregate functions.
- Negations and aggregates can be encoded into rules — SPARQL-query work-around no longer needed.

Goals of the Evaluation

We wanted to Test whether

- RDFox-Datalog rules together with SPARQL queries are expressive enough to compute the HEDIS measures
- RDFox as in-memory triple store can evaluate the data of a whole KP branch on commodity hardware
- RDFox can compute the HEDIS messures in “reasonable” time

End-to-End Evaluation Process

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

Comparison of Results

- The different measures were computed by finding out the patients who satisfied a measure. E.g. 'which diabetic patient had an eye exam?'
- Allowed a patient by patient comparison with the results from the vendor product
- RDFox results could be argued by using RDFox explanations: RDFox gives a proof tree showing how the information is derived.
- RDFox results could therefore be traced back to the raw data.

Advantages of developing with RDFox and RDFox-Datalog

- Legibility of rules uncovers modelling errors earlier: We started out with low discrepancies.
- RDFox-explanations reduce development cycles.
- We even uncovered problems with the vendor solution.
- All RDFox results could be explained

Conclusion and Future work

Conclusion

- We chose a data model described by a standards-based schema ontology which is familiar to domain experts.
- Having a clear model helped a lot during rule authoring.
- RDFox-Datalog + SPARQL queries are enough to express HEDIS CDC
- RDFox computes the results in competitive time on commodity hardware
- Applying Semantic Technologies reduce the development cycles

Future Work

- Semantic Technologies can be applied to encode other regulatory corpora.
- KP consideres the introduction of this technology to compute all of HEDIS.
- Further research projects are envisaged with KP.