Domain Knowledge and Provenance-Integrated Knowledge Organization System Represented with RDFS and SPARQL

Young Soo Song, MD, PhD
Department of Pathology
Division of Informatics
University of Alabama at Birmingham

Agenda

• Model integration in semantic web
• Integration of S3DB domain knowledge and provenance model using SPARQL
  • Current integration status
  • Processes of integration
  • Improved integration status using SPARQL
  • Comparisons SPARQL with JavaScript or Matlab code
• Integrated model as a module (Future work)
Model integration

- Model integration is tedious and difficult.
  - The meaning of natural languages are sometimes unclear.
  - Descriptive power of modeling language is limited.
    - Persistent data is modeled in the early stage → Non-persistent data becomes explicit after coding.
  - Code migration is difficult at the time of integration.
- Semantic web was expected to facilitate the process of data and model integration.
  - Modeling languages: RDFS, OWL
  - Following conventions of semantic web helps integration
    - Triples → syntactic interoperability
    - URI → semantic interoperability
  - Model integration by serialization

Workflow of model integration in semantic web

<table>
<thead>
<tr>
<th>Specification</th>
<th>Semantic web</th>
<th>Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model A</td>
<td>RDF triples</td>
<td>Integrated Model</td>
</tr>
<tr>
<td>Constraints description</td>
<td>a f b.</td>
<td>a f b.</td>
</tr>
<tr>
<td>f(a, b)</td>
<td>c g d.</td>
<td>c g d.</td>
</tr>
<tr>
<td>g(c, d)</td>
<td>a h d.</td>
<td>a h d.</td>
</tr>
<tr>
<td>h(a, d)</td>
<td>a l x.</td>
<td>a l x.</td>
</tr>
<tr>
<td>Model B</td>
<td>RDF triples</td>
<td>b h x.</td>
</tr>
<tr>
<td>Constraints description</td>
<td>i(a, x)</td>
<td>i(a, x)</td>
</tr>
<tr>
<td>j(c, y)</td>
<td>c j y.</td>
<td>c j y.</td>
</tr>
<tr>
<td>k(d, x)</td>
<td>d k x.</td>
<td>d k x.</td>
</tr>
<tr>
<td>Constraints description</td>
<td>h(b, x)</td>
<td>h(b, x)</td>
</tr>
</tbody>
</table>

Model interactions
Limitations of current modeling strategies in semantic web

- Some constraint descriptions are difficult and complex using RDFS or OWL.
  - Abundant class descriptions are supported, but property (predicate) description support is poor.

SPARQL as a model integration tool

- SPARQL can generate RDF triples conforming to constraints of the model.
- Integration can be accomplished through the addition of these triples to the data conforming to existing models.
- More powerful constraints description with SPARQL 1.1
  - Property path
  - Aggregates
  - Subquery
Model integration using SPARQL

Model A
\[ a f b . \\
  c g d . \\
  a h d . \]

Model B
\[ a i x . \\
  c j y . \\
  d k x . \]

Model concatenation
\[ a f b . \\
  c g d . \\
  a h d . \\
  a i x . \\
  c j y . \\
  d k x . \]

Integration

Constraints description
\[ h(b, x) \]

Data conforming to integrated model
\[ a1 f1 b1 . \\
  a2 h1 d1 . \\
  \ldots \\
  c1 j1 y1 . \\
  c2 j2 y2 . \\
  \ldots \]

\[ b1 h1 x1 \rightarrow \text{Query result} \]

S3DB core model

- Knowledge organization system managing domain knowledge, and public and private data.
- Consists of domain knowledge and provenance model.
- Domain knowledge and provenance model is described by RDFS.
- Governance modeling (reacting to provenance) supported by prototype application.
- Incomplete integration by Matlab or JavaScript code.
Current S3DB prototype

Constraints on model interactions

- Model interactions occur in two step processes, propagation and merging.
- Propagation
  - \( \exists (u, f, e, e') (f(u, e) \land D(e, e')) \Rightarrow f(u, e') \)
    where \( u \) is a user, \( f \) is a user operator, \( e \) is an entity,
    \( e' \) is an entity, \( D \) is downstream connected
- Merging
  - \( i = \text{merge}(\{\Phi_A, \phi\}) \rightarrow \begin{cases} 
      & i \mid A = \text{null} = \max(a) \\
      & i \mid A \neq \text{null} = \min(A) 
    \end{cases} \)
    where \( A \) is the vector of indexes of assigned dominant (upper case) states, \( \Phi \); and \( a \) is the index vector of assigned recessive (lower case) states, \( \phi \).
Translation of constraints into SPARQL (Propagation)

PREFIX s3db: <http://s3db.org/ontology#>
PREFIX : <http://s3db.org/provenance#>

Construct
{
  ?u ?o [ s3db:entity ?pe ;
          s3db:entityType "Propagated" ;
          s3db:assignedEntity ?ae ] .
}
WHERE
{
  ?u ?o [ s3db:entity ?ae ;
          s3db:entityType "Assigned" ] .
  ?ae (s3db:DP|s3db:PC|s3db:Rsubject|s3db:Robject|s3db:RP|s3db:CI|s3db:Rpredicate|s3db:Ssubject|s3db:Sobject|s3db:Spredicate)+ ?pe .
}

Relations between a user, a user-operator, and a propagated entity

Translation of constraints into SPARQL (Merging)

PREFIX afn: <http://jena.hpl.hp.com/ARQ/function#>
PREFIX s3db: <http://s3db.org/ontology#>
PREFIX : <http://s3db.org/provenance#>

SELECT ?u ?e (IF(MIN(?la) <= "Z", MIN(?la), MAX(?la)) as ?ans)
WHERE
{
} GROUP BY ?u ?e

Relations between a user, a user-operator, and an entity either assigned or propagated
Propagation and merging as a single query

PREFIX afn: <http://jena.hpl.hp.com/ARQ/function#>
PREFIX s3db: <http://s3db.org/ontology#>
PREFIX : <http://s3db.org/provenance#>

SELECT ?u ?e (IF(MIN(?la) <= "Z", MIN(?la), MAX(?la)) as ?ans) → Merging
WHERE
{
  { ?u ?o [ s3db:entity ?ae ; s3db:entityType "Assigned" ] .
    ?ae s3db:DP|s3db:PC|s3db:Rsubject|s3db:Robject|s3db:PR|s3db:CI|s3db:Rpredicate|s3db:Subject|s3db:Subject|s3db:Spredicate]+ ?e .
  } UNION → Propagation
  { ?u ?o [ s3db:entity ?e ; s3db:entityType "Assigned" ] .
  }
?o : corNum ?la .
} GROUP BY ?u ?e


Comparison between previous code and SPARQL (Propagation)

Matlab code

```
if nargin<3;g=1;end
if nargin<2;n=1;end
nx=length(x);
m1=round(nx/n);m2=ceil(nx/n);mm=m1+m2/n;
if m2==0
    x=char(["\'-ones(1,n)"])
    y=x;
extif mm<1
    y=[x,char(ones(1,n-mx))]
    x=y;
extif mm<1
    y=[x(length(x)-1:length(x))
    end
if g>1
    y=migrate(x,g-1);
```

SPARQL

```
PREFIX s3db: <http://s3db.org/ontology#>
PREFIX : <http://s3db.org/provenance#>

Construct
{
?u ?o [ s3db:entity ?pe ; s3db:entityType "Propagated" ; s3db:assignedEntity ?ae ] .
}
WHERE
{
?u ?o [ s3db:entity ?ae ; s3db:entityType "Assigned" ] .
?ae s3db:DP|s3db:PC|s3db:Rsubject|s3db:Robject|s3db:PR|s3db:CI|s3db:Rpredicate|s3db:Subject|s3db:Subject|s3db:Spredicate]+ ?pe .
}
Comparison between previous code and SPARQL (Merging)

**Matlab code**

```matlab
function merge(x)
if iscell(x)
    x = cell2char(x);
end
if nargin<2;z=unique(lower([x(:)']));end
[n,m]=size(x);
y=char(['-'*ones(1,m)]);
for i=1:m
    x=[lower(x(i));
    x(find(x(i)=='-'))=-1;
    f=find(x==upper(x(i)));
    if ~isempty(f)
        x=x(f);
    end
    if length(x(i))>0
        x=x(i);
        f=find(x==upper(x(i)));
        if ~isempty(f)
            x=x(f);
        end
        if length(x)>0
            xx=lower(x);
            xj=zeros(length(xx));
            for j=1:length(xx)
                xj(j)=find(z==xx(j));
            end
            if xx==x
                y(i)=z(max(xj));
            else
                y(i)=upper(z(min(xj)));
            end
        else
            var = new Array;
            var = new String;
            y = new String;
            for j=1:length(xx)
                y(j)=null;
            end
        end
    else
        var = new Array;
        var = new String;
        y = new String;
        for j=1:length(xx)
            y(j)=null;
        end
    end
end
end
```

**SPARQL**

```sparql
PREFIX afn: <http://jena.hpl.hp.com/ARQ/function#>
PREFIX s3db: <http://s3db.org/ontology#>
PREFIX : <http://s3db.org/provenance#>

SELECT ?u ?e (IF(MIN(?la) <= "Z", MIN(?la), MAX(?la)) as ?ans)
WHERE {
    ?o corLabel ?la
} GROUP BY ?u ?e
```

60% of the total
Could an integrated model be used as a module? (Future works)

- RDF representation of SPARQL is required.
- SPIN
  - A standards-based way to define rules and constraints for semantic web data
  - Business rules expressed in SPARQL are represented as RDF

\[ f(g(x)), \quad \text{where} \quad f : \text{External SPARQL}, \quad g : \text{SPIN-represented SPARQL}, \quad x : \text{RDF data} \]

Conclusions

- We presented a mechanism to complete the integration of models using SPARQL.
- S3DB prototype has a more simplified and adaptable structure, enabling more efficient management and computation.
- RDF representation of SPARQL using SPIN may make a modular representation of the integrated model possible.