RDF Browser for Data Discovery and Visual Query Building

Jans Aasman
Franz Inc.
2201 Broadway, Suite 715, Oakland, CA 94612
ja@franz.com

Ken Cheetham
Franz Inc.
2201 Broadway, Suite 715, Oakland, CA 94612
cheetham@franz.com

ABSTRACT
The free-form nature of triplestores offers a lot of flexibility for constructing databases, but that freedom can also make it less obvious how to find arbitrary data for retrieval, error-checking, or general browsing. Gruff is a graphical triplestore browser that attempts to make data retrieval more pleasant and powerful with a variety of tools for laying out cyclical graphs, displaying tables of properties, managing queries, and building SPARQL and queries as visual diagrams.

Author Keywords
RDF, SPARQL, Query Builder, Triplestore, Prolog, Reasoning, Semantic Web

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

There is an exciting explosion of linked RDF datasets being made available in the Linked Open Data Cloud ("LODC"). A current overview is described in Bizer et al's paper, Linked Data - *The Story So Far.* [1]. The Authors argued that we have moved beyond the point where we simply publish some info as RDF and contend that we have arrived at early maturity: there are now quality standards, best practices and a whole slew of tools to help with publishing and retrieving RDF.

Many RDF datasets from the LODC are about one particular domain and contain a description of the classes (the ontology), a set of instances, and possibly some explicit owl:sameAs relations to other instances in other datasets. Some other RDF datasets in the LODC actually have many different domains, DBPedia and Freebase are good examples. The LODC vision is that the single data representation for every type of knowledge (RDF) will make it easy to combine datasets and dramatically increase their value. In practice, the exploitation and exploration of these data sources can be far from trivial. Two papers recently made this point:

1. Heath [2] argues in his paper *How will we interact with the web of data?* the tools that are currently being developed basically focus on how machines can interact with the RDF cloud. However, there is also an urgent need for new Human-Computer interaction paradigm for the Web of Data. One point he makes, it is actually better to speak of a Web of Things and visualization tools should make the thing that is referred to in RDF become a first class object.

2. Karger and Schraefel [3] wrote an interesting position paper called *The Pathetic Fallacy of RDF*. The core of their criticism of current visualization tools is that the cloud of data is treated as one big fat graph that somehow should be displayed as a graph. Their paper shows that in many cases the 'graph view' hardly helps the user in achieving their tasks. We actually agree with their assessment and we think that any visualization tool should be a combination of graphs, table like views, object oriented views and faceted navigation and browsing tools.

But lets us go back to why it is hard to browse the LOD cloud. To begin, most of the time an interesting problem requires you to combine datasets. A typical example would be from the biomedical domain. Say you take the RDF version of Clinical Trials [4], the side effect database – Sider [5], the disease database - Diseasome [6], Dailymed [7] with commercially available medicine, and Drugbank [8] filled with FDA approved drugs and targets. With all these databases combined you can perform some very interesting analysis to find relationships between clinical trials and genes, drug and diseases in a space of more than a 100 different classes and thousands of types of relationships.

EXPLORATION AND DISCOVERY

So what kinds of analysis would you do with these datasets:

Workshop on Visual Interfaces to the Social and Semantic Web (VISSW2011)
Co-located with ACM IUI 2011
Feb 13, 2011, Palo Alto, US.

Example 1 - Automatic Discovery. Sometimes users just want to know if it would be possible to find any connection between two interesting instances from two different datasets. A good example would be to find all the links you can find between a particular target (Cytochrome c3) and say Morphine Sulfate given all the predicates or a subset of the predicates in the database. What is non-trivial about this? Well most Triplestores do not provide automatic discovery functions that would connect these instances or provide ways to specify which predicates to use while doing automatic discovery.

Example 2 - Write New Queries. Say a user has two instances from two different datasets (like Example 1) and you want to write a SPARQL query that links them together. Again, what is non trivial about this? In some sense a user has to know the datasets before they can write the queries, that is, you have to understand the predicates and the names of the classes that will link them together.

Unfortunately the datasets from the example above lack domain and range restrictions so it not straight forward to explore the schema space. Additional usability problems are that you have to know the namespaces, the fragments of the namespaces, and very often the type of the objects.

Example 3 - Basic Data Mining: Say you have an instance of a clinical trial and now you want to find all the clinical trials that discuss the same kind of drugs, diseases and targets. The non trivial part is that most triple stores do not offer you advanced ways to perform stored procedures on large datasets that could do this data mining for you.

In order to make exploration of multiple datasets easier we developed Gruff, an advanced Graphical User Interface for working with RDF Data in a triplestore.

Gruff provides a full set of graph analysis capabilities to help users explore the LODC. In the most simple case a user selects two resources in Gruff, the predicates the graph algorithm needs to explore, and Gruff will find all connections between the two instances. The end result on the screen can even be saved as an NTriple file. The graph algorithms all take as input first class functions called generators. The contract for a generator is that it takes one node as an input, and will generate a set of nodes as output. Gruff will create these generators for you if you select the predicates that you want to explore.

In other cases the users want to investigate something more sophisticated, like conditionally exploring predicates per type of node (or resource), starting with the important predicates first. In this case the user needs to write some SPARQL or Prolog code and this can be facilitated with the visual query builder described below.

With Gruff you can first explore the graph over multiple link data sets completely visually, and once you find one pattern that you are interested in, you can easily build from that a visual query and find all the same patterns in the set of linked datasets.

Gruff provides several key capabilities for LODC exploration:

Graph Layout:

Triplestores specialize in relationships between things, and it can be difficult to get a feel for a tangled set of relationships.

Gruff's Graph View (figure 1) provides automatic layout of highly-cyclic graphs. Heuristics are used to make layouts of a modest number of nodes fast enough for interactive browsing. The algorithm specializes in keeping nodes from laying on top of unrelated link lines, even though all link lines are straight; those two features allow you to quickly spot all of the nodes that are linked with a particular node, and then to quickly spot all of the linked nodes of those nodes, and so on.

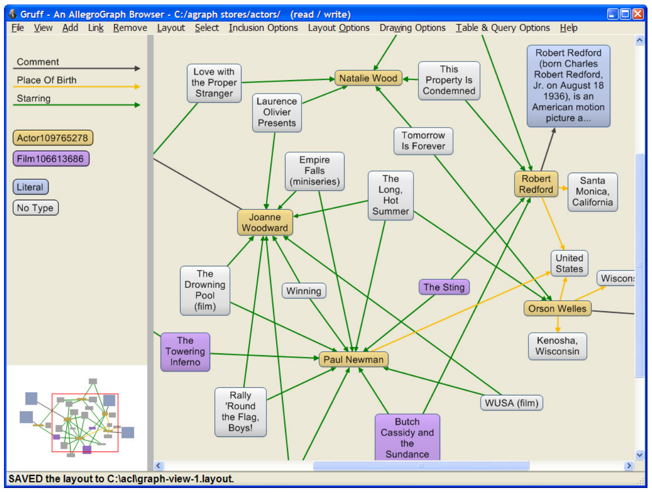
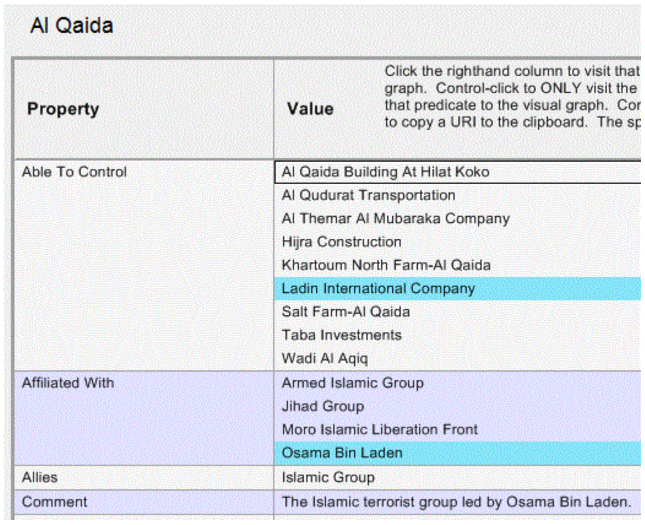

Gruff - An AllegroGraph Browser - C:/agraph stores/actors/ (read / write)
File View Add Link Remove Layout Select Inclusion Options Layout Options Drawing Options Table & Query Options Help
Comment
Place Of Birth
Starring
Actor109765278
Film106613686
Literal
No Type
Love with the Proper Stranger
Natalie Wood
This Property Is Condemned
Robert Redford (born Charles Robert Redford, Jr. on August 18 1936), is an American motion picture a...
Laurence Olivier Presents
Tomorrow Is Forever
Empire Falls (miniseries)
The Long, Hot Summer
Robert Redford
Santa Monica, California
Joanne Woodward
United States
Wiscons
The Drowning Pool (film)
Winning
The Sting
Orson Welles
Kenosha, Wisconsin
The Towering Inferno
Paul Newman
Rally 'Round the Flag, Boys!
WUSA (film)
Butch Cassidy and the Sundance
SAVED the layout to C:\acl\graph-view-1.layout.

Figure 1.

Once you have found one or more initial objects of interest, the familiar Table View (figure 2) can be used to jump to linked objects that are properties of other objects. The Graph View will add nodes and links that show a history of all of the paths that you traversed in the Table View. You can also incrementally add linked nodes directly in the Graph View.



Al Qaida

Property	Value	Click the righthand column to visit that graph. Control-click to ONLY visit the that predicate to the visual graph. Cor to copy a URI to the clipboard. The sp
Able To Control	Al Qaida Building At Hilat Koko	
	Al Qudurat Transportation	
	Al Themar Al Mubaraka Company	
	Hijra Construction	
	Khartoum North Farm-Al Qaida	
	Ladin International Company	
	Salt Farm-Al Qaida	
	Taba Investments	
	Wadi Al Aqiq	
Affiliated With	Armed Islamic Group	
	Jihad Group	
	Moro Islamic Liberation Front	
	Osama Bin Laden	
Allies	Islamic Group	
Comment	The Islamic terrorist group led by Osama Bin Laden.	

Figure 2.

Finding arbitrary objects:

A triplestore can contain many objects that are not instances of a simple set of classes, and so it can be tricky to locate arbitrary objects of interest.

Gruff has a variety of ways for looking up arbitrary objects. These include (1) specifying an object textually as either its easy-to-type rdfs:label string, its full URI, or a brief URI that uses a namespace identifier; (2) selecting a node that has a selected rdf:type (selecting the type either from a single list or from an rdfs:subClassOf hierarchy); (3) finding all nodes that have property values containing arbitrary text, using free text lookup; and even (4) selecting from all sorted nodes.

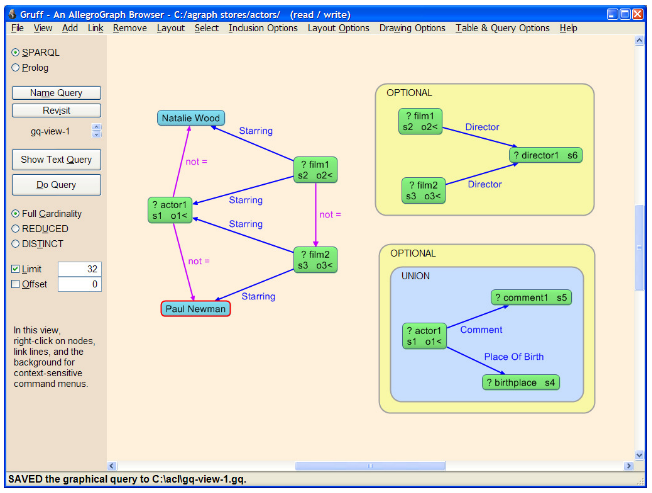
When there are many choices to select from, Gruff manages the selection by using a series of pop-up menus that allow you to gradually hone in on a desired object. You use these menus by first selecting the starting character of an object's name, then the second character, and so on. Each menu displays only the choices that remain for the next character, with string completion for the remaining choices to the extent that's possible without ambiguity, allowing you to see increasingly recognizable choices.

Saving visual layouts and other views to file allows you to start back up quickly in a future Gruff session with a familiar set of "starter nodes", rather than specifying them again from scratch. The most recently saved or loaded views are always listed on the View menu for quick retrieval. You can also copy nodes from one view to another, for example, to avoid respecifying them.

Building Queries Graphically

While query languages like SPARQL are quite powerful, queries can become rather complex and therefore difficult to construct free of nesting mismatches and typos in URIs.

Gruff's newest feature, the Graphical Query View (figure 3), allows creating queries as diagrams of nodes and links. A query diagram can include actual objects from the store, which you select as in other views, while other nodes and links represent query variables. Group graph patterns such as UNION and OPTIONAL groups can be laid out as grouper boxes that can be nested to any level, with proper nesting maintained automatically.

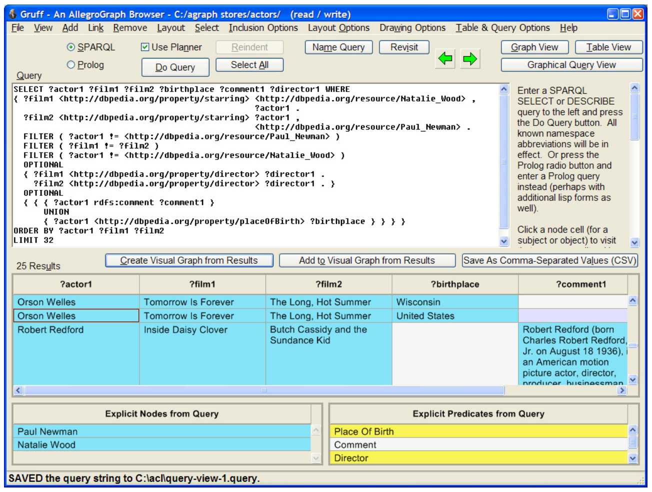


In this view, right-click on nodes, link lines, and the background for context-sensitive command menus.

SAVED the graphical query to C:\acl\gq-view-1.gq.

Figure 3.

Once you've created a query diagram, Gruff will generate either SPARQL or Prolog code for the query. Seeing the generated query text (figure 4) can help you learn to write SPARQL or Prolog queries directly. You can also edit the generated text before performing the query. You can save queries either as graphical diagrams or as actual query text and load them any time later for modification into related queries. You can even generate a visual graph from query results, showing the network of all triples that are comprised of nodes and predicates that are in the query results and the query itself.


Gruff - An AllegroGraph Browser - C:/agraph stores/actors/ (read / write)
File View Add Link Remove Layout Select Inclusion Options Layout Options Drawing Options Table & Query Options Help
SPARQL
Prolog
Use Planner
Reindent
Do Query
Select All
Name Query
Revisit
Graph View
Table View
Graphical Query View
Query
Create Visual Graph from Results
Add to Visual Graph from Results
Save As Comma-Separated Values (CSV)

```
SELECT ?actor1 ?film1 ?film2 ?birthplace ?comment1 ?director1 WHERE
{ ?film1 <http://dbpedia.org/property/starring> <http://dbpedia.org/resource/Natalie_Wood> ,
                                                ?actor1 .
  ?film2 <http://dbpedia.org/property/starring> ?actor1 ,
                                                <http://dbpedia.org/resource/Paul_Newman> .
  FILTER ( ?actor1 != <http://dbpedia.org/resource/Paul_Newman> )
  FILTER ( ?film1 != ?film2 )
  FILTER ( ?actor1 != <http://dbpedia.org/resource/Natalie_Wood> )
  OPTIONAL
  { ?film1 <http://dbpedia.org/property/director> ?director1 .
    ?film2 <http://dbpedia.org/property/director> ?director1 . }
  OPTIONAL
  { { { ?actor1 rdfs:comment ?comment1 }
      UNION
      { ?actor1 <http://dbpedia.org/property/placeOfBirth> ?birthplace } } } }
ORDER BY ?actor1 ?film1 ?film2
LIMIT 32
```

Enter a SPARQL SELECT or DESCRIBE query to the left and press the Do Query button. All known namespace abbreviations will be in effect. Or press the Prolog radio button and enter a Prolog query instead (perhaps with additional lisp forms as well).

Click a node cell (for a subject or object) to visit

25 Results

?actor1	?film1	?film2	?birthplace	?comment1
Orson Welles	Tomorrow Is Forever	The Long, Hot Summer	Wisconsin	
Orson Welles	Tomorrow Is Forever	The Long, Hot Summer	United States	
Robert Redford	Inside Daisy Clover	Butch Cassidy and the Sundance Kid		Robert Redford (born Charles Robert Redford, Jr. on August 18 1936), i an American motion picture actor, director, producer, businessman

Explicit Nodes from Query
Paul Newman
Natalie Wood

Explicit Predicates from Query
Place Of Birth
Comment
Director

SAVED the query string to C:\acl\query-view-1.query.

Figure 4.

CONCLUSION

In general, Gruff's tools are tightly integrated to allow browsing a triple-store in a variety of ways. Gruff is free and can be downloaded from www.franz.com/agraph/gruff/, where additional screen shots and demonstration videos can also be found.

REFERENCES

1. Heath, T., Hepp, M., and Bizer, C. *The Story So Far*. Special Issue on Linked Data, International Journal on Semantic Web and Information Systems (IJSWIS) 2009. http://linkeddata.org/docs/ijswis-special-issue
2. Tom Heath (2008) *How Will We Interact with the Web of Data?* IEEE Internet Computing, Vol. 12(5), pp. 88-91
3. Schraefel, M. and Karger, D. (2006) *The Pathetic Fallacy of RDF*. In: International Workshop on the Semantic Web and User Interaction (SWUI) 2006, Nov 2007, Athens, Georgia.
4. LinkedCT http://linkedct.org/index.html
5. Sider - http://sideeffects.embl.de/
6. Diseasome - http://www.nd.edu/~alb/Publication06/145-HumanDisease_PNAS-14My07-Proc/Suppl/.
7. DailyMed - http://dailymed.nlm.nih.gov/
8. Drugbank - http://www.drugbank.ca/