

Gruff QuickStart

Contents

- Gruff – Intro to key features
 - a visualization tool for inspecting Semantic Graph Databases (triple stores) *and* building SPARQL/Prolog Queries visually
- Example Data - <https://allegrograph.com/downloads/>
 - Healthcare Data
 - Movie Actors Data
 - Crunchbase Companies
- Quickstart lessons in Gruff

What version of Gruff?

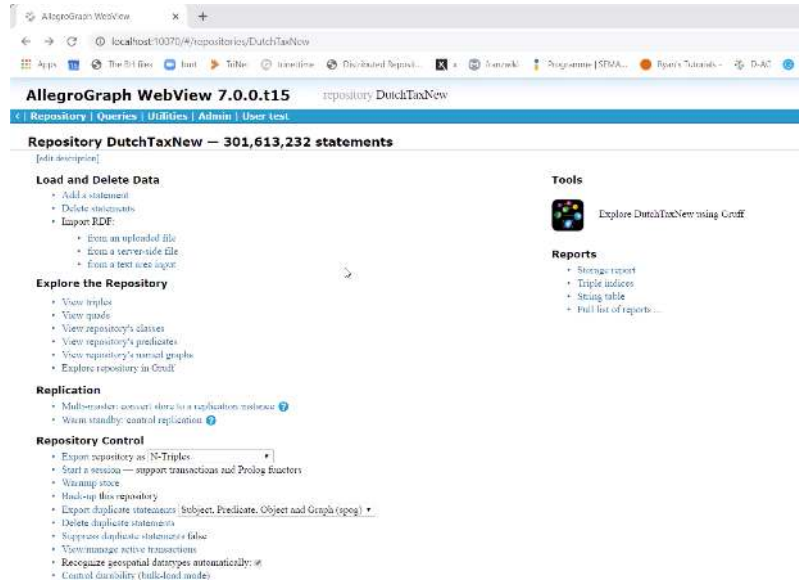
- Gruff v8.0 for AllegroGraph 7 – **Free** up to 5 M triples
 - New in v8 – Browser based – Integrated with AllegroGraph AGWebView.
 - For large datasets and production systems – info@franz.com
- Gruff actually works with ANY triple store via SPARQL Endpoint Protocol.
- Demo Version – No Download required
 - <http://gruff.allegrograph.com:10035/#>

[1] Install AllegroGraph – Gruff Built-in

- Install and Start AllegroGraph
- Download and Load actors.ntriples.gz
- Open the 'actors' repository
- Click on the Gruff icon
- Menu>Display>Display Some Sample Triples
- Remove all triples from screen ('z' till blank, or ctr-alt-x)

- Create free text index (Text Search>Select or Create Text Index) and select all the predicates you want to include
- Search for a movie actor using text search ('h')

[2] Manage your Repositories with AGWebview



- Gruff - Now Integrated with AGWebView
- Browse available catalogs and repositories
- Create (also delete) a repository
- Load RDF data into a repository, or into a specific graph of a repository.
- Configure triple indices for a repository
- Set up free-text indexing for a repository
- Issue SPARQL and Prolog queries, which can be saved and reused
- Walk from query results to related triples and resources
- View and add namespaces
- Manage AllegroGraph users and “roles,” and their access to repositories
- Capture a query as a web URL for embedding in applications
- Apply Prolog rules and functors to repositories
- Open sessions for commit and rollback.
- Much more

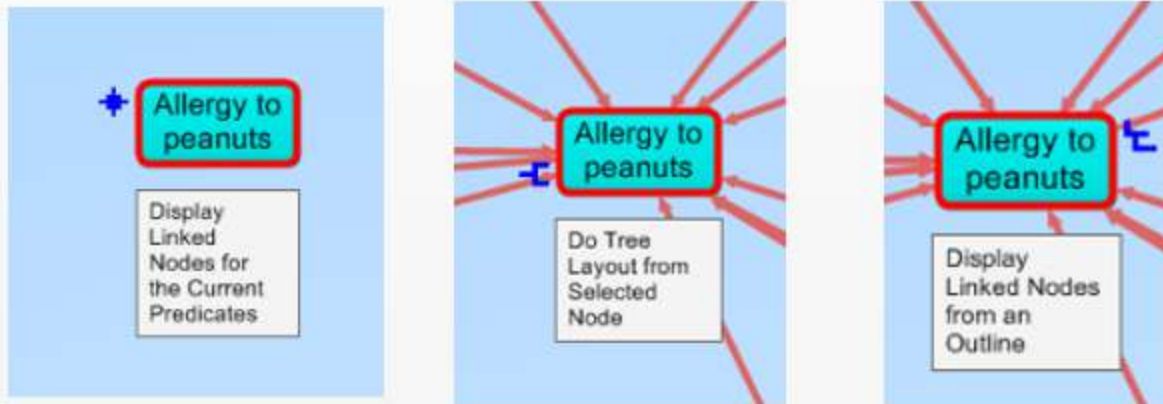
[3] Graph View

- Most time spent in the Graph view (press g)
- Some easy starters to get something on the screen
 - Using text indexing (press h)
 - Do simple SPARQL query in query screen (press w) ..
 - `Select * where { ?x a ?y . } limit 10 ..` Click on 'create-visual-graph'
 - Display sample instance node by class hierarchy (press ctr-j)
- Select the predicates to show on the screen (press p)
- Use shift-p to find an earlier selection
- Explode nodes (press f)
- Going back to previous state (press z)
- Reorder graph (press r)
- Center the graph (press c)
- Make the graph more compact (press 6)
- Nudge the graph (press d)
- Zoom in and out (middle wheel mouse, or use , and .)

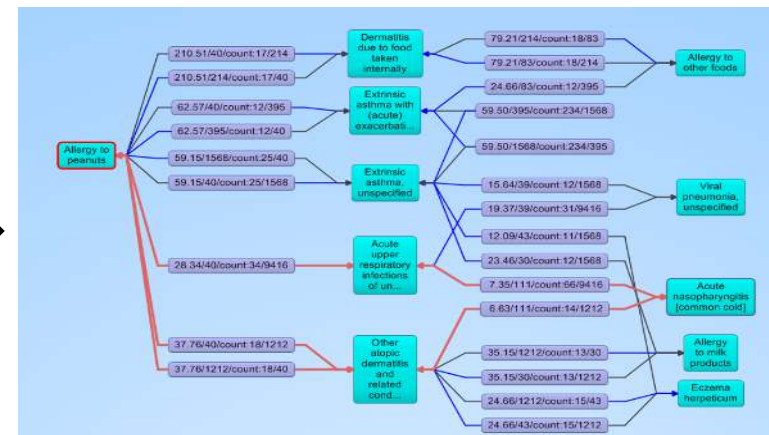
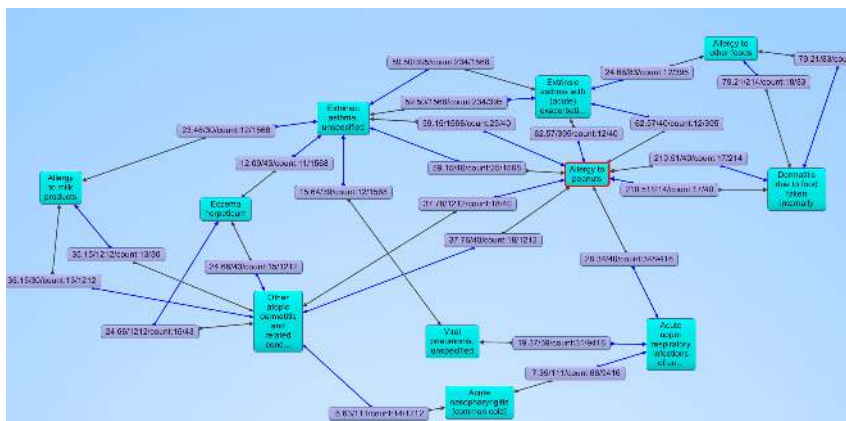


[3] Graph View Layouts

- The corners of nodes are 'mouse-sensitive': please try all corners



- And try our tree rendering using ctr-t (or left bottom node corner)



[3] Graph View Continued

- Explode node by right-clicking and choosing from a menu or a tree
- Select and then delete nodes from the screen (press x)
- Delete all nodes from the screen (press ctr-alt-x)
- Eraser mode (press ctr-shift-x)
- Remove orphans (press shift-o)
- Show full namespaces (shift-8 .. **Not F8**)
- Toggle predicate names on screen (press n)

- Find shortest path (press shift-f)
 - Pin nodes, or graph might change too much (press i)
 - Find shortest path with shift-f
 - What if you can't find a path?
 - Choose more predicates and sometimes less ☺
 - Create larger time window to search
 - Buy more memory and solid state drives for your laptop
 - Check in table view if one of the nodes has the right predicates

[3] Graph View 'G' continued

- Sometimes you don't want to see the node labels (ctr-8 or
 - Visual-graph-options -> node labels -> use label preds for nodes)
- Important: you can add your own node labels..
- You can always see the full URL of a node left under the screen

- Double click on node to go to table view (or press t)
- Go directly to outline view from selected node (press o)

- Save a graph layout to disk (ctr-s)
- Load a graph layout (ctr-l)
- Graph layouts are simple text files and can be shared

[4] Table View

IF YOU WANT TO SEE YOUR TRIPLES IN A TRADITIONAL TEXTUAL TABLE (press t or double click a node)

- Navigate the data by clicking on 'values'
- Above gray line: s p o (where s is the main node we selected)
- Under gray line: o p s (note the 'is' predicate 'of) to denote the other direction
- Note how '8' shows you full URLs of nodes
- Note how 'Alt-8' shows you how we use node-labels
- You can create triples in the table view, see manual...

[5] Outline View – press o

IF YOU HAVE DEEP HIERARCHICAL DATA AND YOU NEED TO RETAIN THE CONTEXT WHILE NAVIGATING.

- Navigate the data by clicking on little arrows
 - Blue: outgoing triples
 - Black: triples pointing to previous level (note the ‘is’ ... ‘of’ around the predicates
- Display:
 - Note how ‘8’ shows you full URLs of nodes
 - Note how ‘Alt-8’ shows you how we use node-labels



[6] Query View – press w

Write queries in SPARQL and Prolog – by writing code

- Use namespace abbreviations (b) to avoid typing prefixes
- Turn output into Graph or CSV
- Managing queries
 - Save queries (ctr-s)
 - Load queries (ctr-l)
- Investigate menu ... query options
 - One can change the time limit..

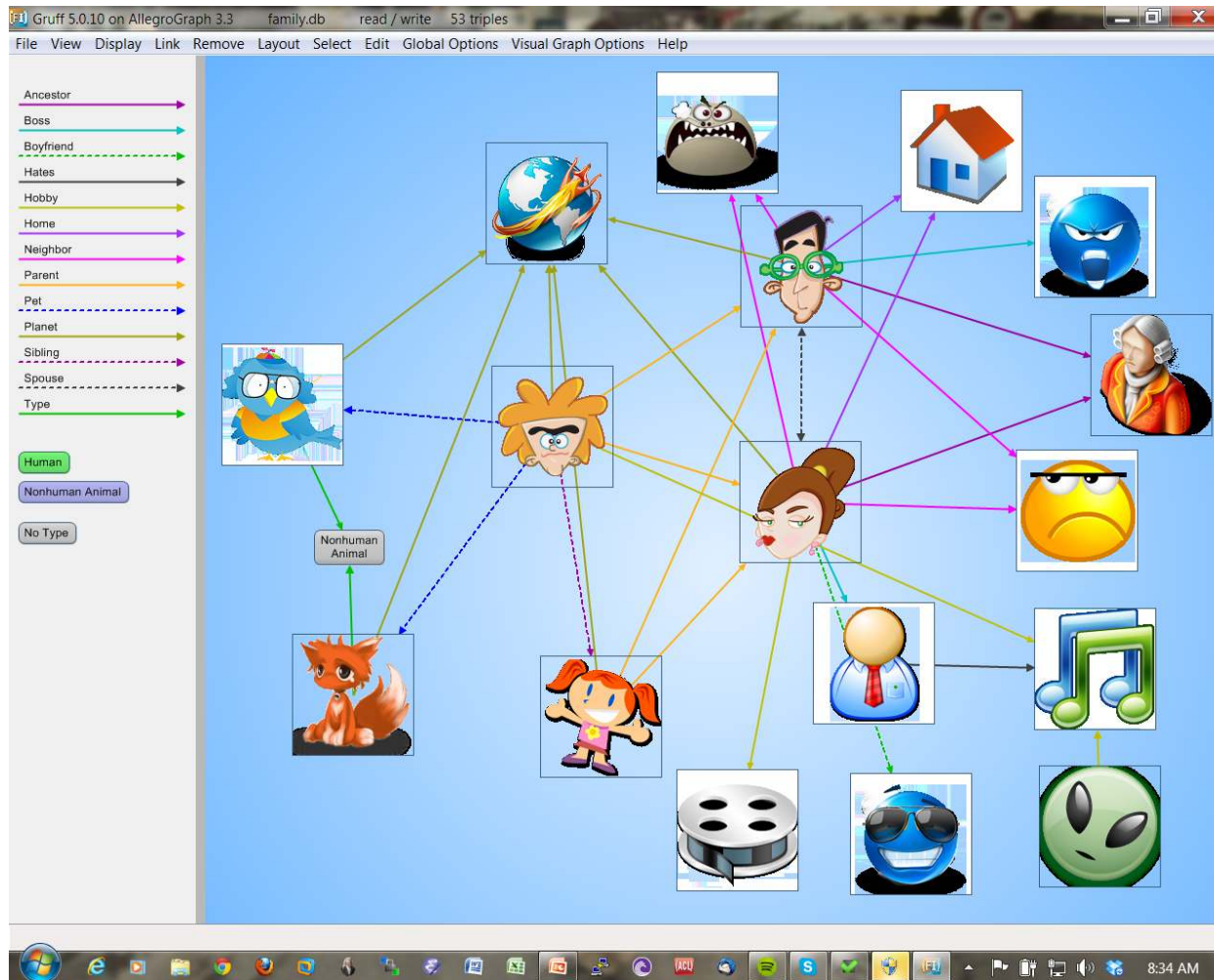
[7] Visual Query Editor – press e

- Build queries visually – no code!
- Most things work like the graphical query view
 - Search for freetext (press h)
- Right click on screen to create new variables
- Right click on variables to create new predicates with drag and drop
 - Multiple ways to choose names of predicates.. See menus

[8] Discovery: using graph patterns on the screen to build queries

- Create interesting patterns in the Graph view using everything you learned so far
- Find a connection trail that you think is interesting
- Highlight each node in trail (ctr-left-click or 'h')
- Move highlighted nodes to query editor (ctr-alt-i)
- Change some variables manually or let Gruff find the variable names for you

[9] Using Pictures for Nodes



[9] Using Pictures for Nodes

- Tell Gruff what predicate to use for pixmaps
 - `<jans> <pixmap> "pictureOfJans.ico"`
 - Global-options -> node label predicates -> custom predicates for node label pixmaps.
- And tell Gruff the base location of your pixmaps
 - You can use relative filenames
 - You can use `jpg png bmp ico`
 - Global-options->miscellaneous->document base folder