WQL/WSI web interface for querying profiles

Alexandre Rademaker, IBM Research and FGV/EMAp

WeSearch

- The WeSearch Project
  - https://github.com/delph-in/docs/wiki/ErgWeSearch
  - https://github.com/delph-in/docs/wiki/WeSearch_QueryLanguage
  - https://github.com/delph-in/docs/wiki/WeSearch_Rdf
  - http://moin.delph-in.net/wiki/WeSearch/Interface (not migrated)
  - Many pages about WeSearch were not migrated

WeSearch

http://wesearch.delph-in.net/

Built using Java Enterprise technologies (Apache Jena, Apache Tomcat, and others), hence near-trivial to maintain and adapt for adult software engineers. (https://github.com/delph-in/docs/wiki/VirtualInfrastructure#wesearch-semantic-graph-query)

http://sdp.delph-in.net/

http://sdp.delph-in.net/2014/ (SemEval 2014 Task 8)

Semantic fingerprints (vs WQL)

Some of the ERG treebank data can be searched via semantic ‘fingerprints’ using the WeSearch Interface; for information on fingerprinting, please see the ErgSemantics documentation.

https://github.com/delph-in/docs/wiki/ErgTreebanks

https://github.com/delph-in/docs/wiki/ErgSemantics#semantic-fingerprints

> In capturing semantic phenomena on most **ESD** pages (and hopefully also in future work on automated regression testing) we invoke a notion of semantic fingerprints, i.e. characteristics of a specific MRS configuration that identifies a token phenomenon. We utilize a compact template language for MRS fingerprints (similar in form to the MRS LaTeX style; called ERS fingerprints when specialized for the semantic analyses of the English Resource Grammar)
>
> “ErgSemantics fingerprint language is the WQL dialect for MRS search. i am afraid, i believe no documentation is available for this dialect.” (Stephan Oepen)

The WSI system

Hosted at IBM Cloud

https://wsi.mybluemix.net/demo/search.jsp

Documentation:

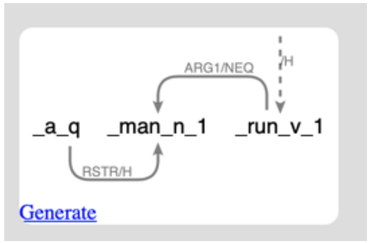
http://sdp.delph-in.net/2015/search.html

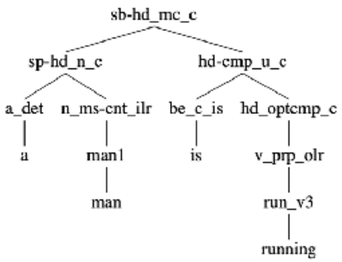
Code at:

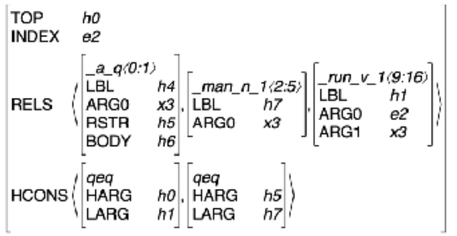
https://github.com/own-pt/wsi

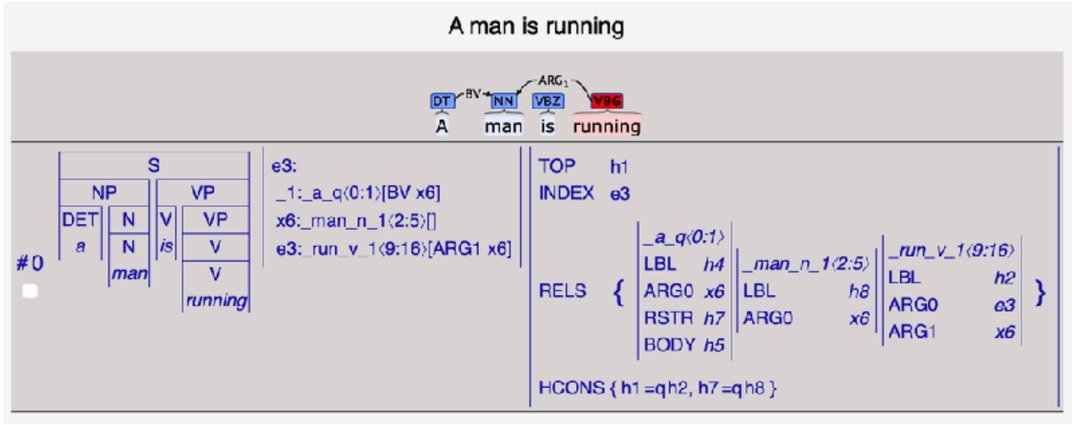
Web visualization

- http://delph-in.github.io/delphin-viz/demo/ (JS + SVG + CSS)
- http://chimpanzee.ling.washington.edu/demophin/erg/parse (DMRS)
- http://erg.delph-in.net/logon (DM, MRS AVM-Like, EDS) (Lisp + HTML + CSS)
- https://wsi.mybluemix.net/demo/search.jsp (JS + Java + HTML + CSS)
- http://compling.hss.ntu.edu.sg/ltdb/cgi/ERG_1214//showtype.cgi?typ=aj_-_i-att_le



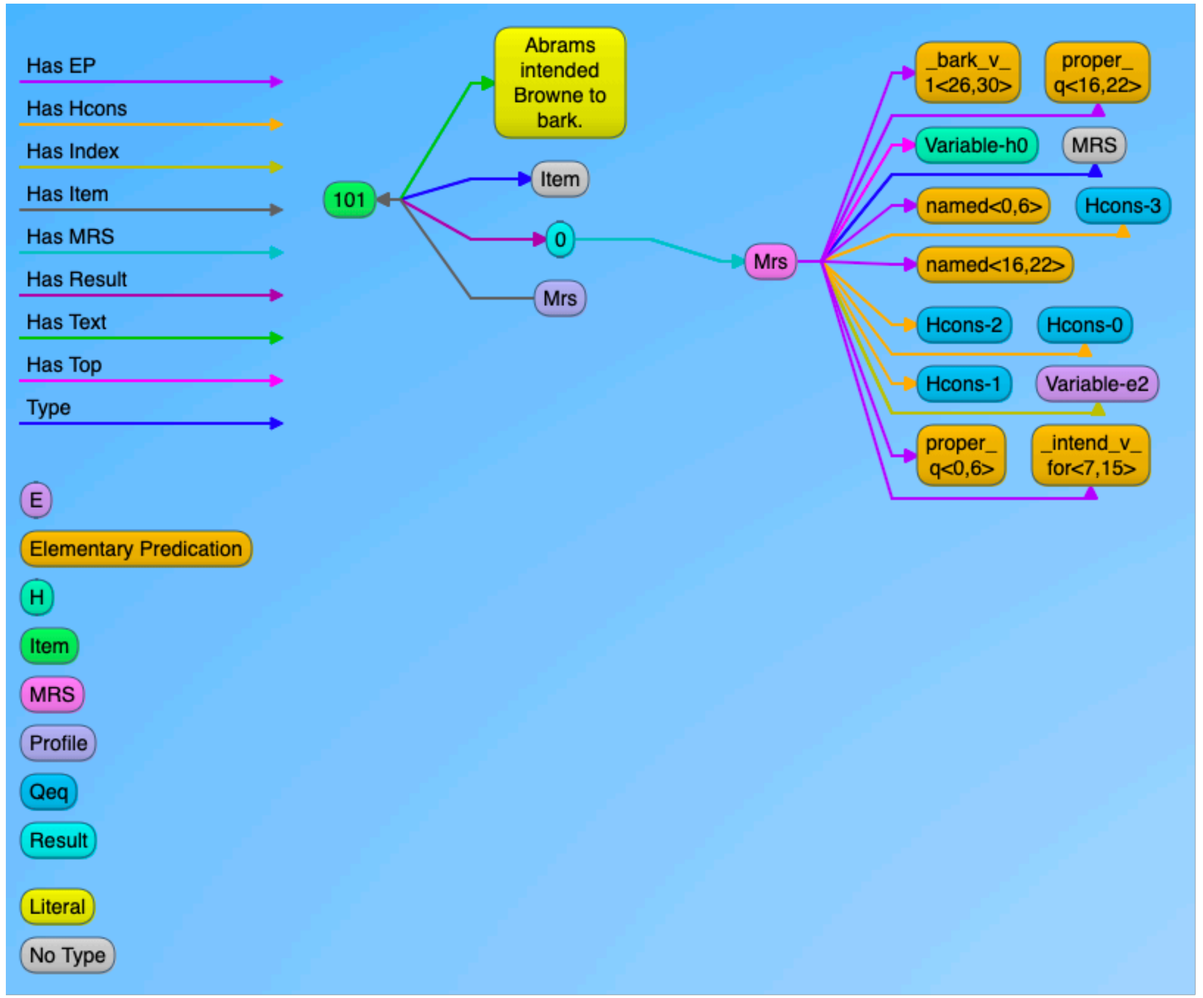






RDF Representation

- https://github.com/own-pt/delphin-rdf/tree/master/vocabularies
  - erg.ttl
  - semstructs.ttl
    - eds.ttl
    - dmrs.ttl
    - mrs.ttl
- URI pattern
  - http://example.com/
    - item/result/mrs#…
    - Item/result/dmrs#…



The new system

Haskell

https://hackage.haskell.org/package/hsparql

https://github.com/own-pt/wql

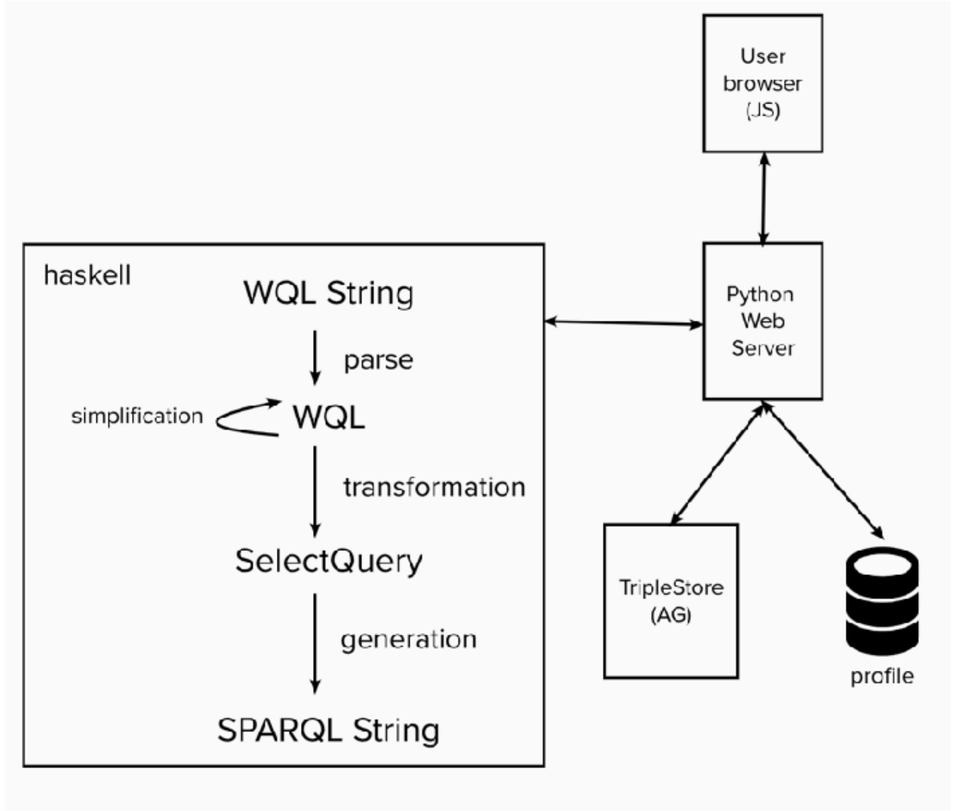
Python

Pydelphin + delphin-rdf plugin

https://github.com/own-pt/delphin-rdf

JavaScript

https://github.com/delph-in/delphin-viz



Questions and TODO

- WQL
  - variables (x vs ?x) vs patterns
  - MRS, DMRS, EDS dialects?
- visualization:
  - highlighting matches
  - show derivations, syntactic tree etc
- SPARQL pretty printing

Step by Step

setup TripleStore

clone and create a virtual env with Pydelphin + delphin-rdf

delphin profile-to-rdf -o mrs.nq -f nquads -p http://www.delph-in.net/data/erg/gold/mrs ~/hpsg/terg/tsdb/gold/mrs

import RDF; anonymous access to the store in AG

cd wql ; stack build ; stack exec wql-exe

FLASK_APP=demo flask run

Visit http://127.0.0.1:5000/test-form