

Ontology View Query Management

Landon T Detwiler¹, MS, Marianne Shaw², MS, James F Brinkley^{1,2,3}, MD, PhD
Departments of Biological Structure¹, Computer Science and Engineering², Medical
Education and Biomedical Informatics³, University of Washington, Seattle, WA

Like views in relational databases, ontology views are expressed as queries, but over source ontologies rather than tables. To enhance the reusability of such views, we are constructing a view Query Manager application. The Query Manager allows queries to be edited, executed, and stored for reuse. View queries are discoverable by searching the Query Manager's metadata catalog. The Query Manager also supports the storage of materialized view results upon which further queries may be issued.

Knowledge driven applications often require application specific derivatives of existing ontologies. When such derivatives are expressed via a query, conceptually a virtual graph composed of the query results, we refer to this as an ontology view. A view serves not only as the definition of a virtual graph but also as its provenance record.

The web based Query Manager (QM) provides an interface and framework for editing, executing, storing, and discovering views. At present it supports queries in SPARQL, vSPARQL [1], XQuery, or DXQuery [2] (where vSPARQL and DXQuery are extensions of SPARQL and XQuery respectively).

While a view represents a virtual graph, applications often require a more concrete artifact. View queries may be executed (materialized) by the QM and the results stored locally, to the users machine, or archived within the QM. Locally stored results may be embedded into knowledge driven

applications. These derivatives are updatable, on demand, by re-executing their associated view queries. Results stored within the QM are made available as new sources to query.

The QM web client is implemented in Adobe Flex and communicates with a Java-based server via a remote object service. Queries and materialized views, along with their associated metadata are stored in a relational database. The QM server does not directly execute queries. Instead query processing is allocated to external services. At present, SPARQL and vSPARQL queries are processed by the service at [3] and XQuery and DXQuery queries are processed by [4].

A running demo of the Query Manager can be accessed at [5]. Future work includes: integration with the BioPortal ontology repository [6], inclusion of a graphical interface for query editing (or other non-graphical query construction assistance), aid in the discovery of resources for querying (ontologies or views), and the ability to query over Linked Data.

Supported by NIH grant HL087706

[1] <http://sigpubs.biostr.washington.edu/archive/00000227/>

[2] <http://sigpubs.biostr.washington.edu/archive/00000234/>

[3] http://axon.biostr.washington.edu:8080/VSParQL_Service/services/VSParQLService?wsdl

[4] http://axon.biostr.washington.edu:8080/DXQP_Service/services/DXQP_ws?wsdl

[5] <http://axon.biostr.washington.edu:8080/QueryManager/QueryManager.html>

[6] <http://bioportal.bioontology.org/>

Structural Informatics Group

Query Manager

Hello Todd Detwiler [New Query](#) [Search](#) [Register](#) [Login](#)

Title
Continuities of the pharynx

Description
Select all of the structures directly continuous with the pharynx according to the FMA

Query

```
PREFIX fma: <http://sig.biostr.washington.edu/fma3.0#>
SELECT ?pharynx_continuity
FROM <http://sig.biostr.washington.edu/fma3.0>
WHERE
{
  fma:Pharynx fma:continuous_with ?pharynx_continuity .
}
```

[Execute](#) [Save](#) Language [VSparQL](#)