

A Declarative Query Interface for Large Semantic Networks

Peter Mork^{2,3}, Dan Suciu²,
Jim Brinkley^{1,3}, Cornelius Rosse¹

¹Structural Informatics Group, ²Computer Science & Engineering, ³Biomedical & Health Informatics; University of Washington

Abstract

The OQAFMA Query Agent for the Foundational Model of Anatomy (OQFMA) was designed to enhance the capabilities of an earlier server and to provide more rapid access to a large knowledge base. The system architecture was optimized using indexes for the most common classes of queries. Additional performance benefits were gained by pre-computing the transitive closure of appropriate relationships. Flexible, random access of the data is provided via a declarative query language, derived from StruQL, a language for querying semi-structured data.

Background/Related Work

The Foundational Model of Anatomy¹ (FMA) is a semantically expressive ontology of anatomical entities that constitute the human body. It consists of more than 100,000 concepts, which are connected by more than 1 million relationships. Given its size and complexity, efficient access is important.

Currently two systems provide access to the FMA: The Foundational Model Server² (FMS) and Protégé-2000³. The FMS uses a remote procedure call architecture that supports a pre-specified collection of LISP-like function calls. Given its limited scope, rapid access is supported. The Protégé-2000 API (which is closely related to OKBC⁴) provides a greater degree of flexibility in interacting with the database, but this flexibility comes at the cost of speed and complexity.

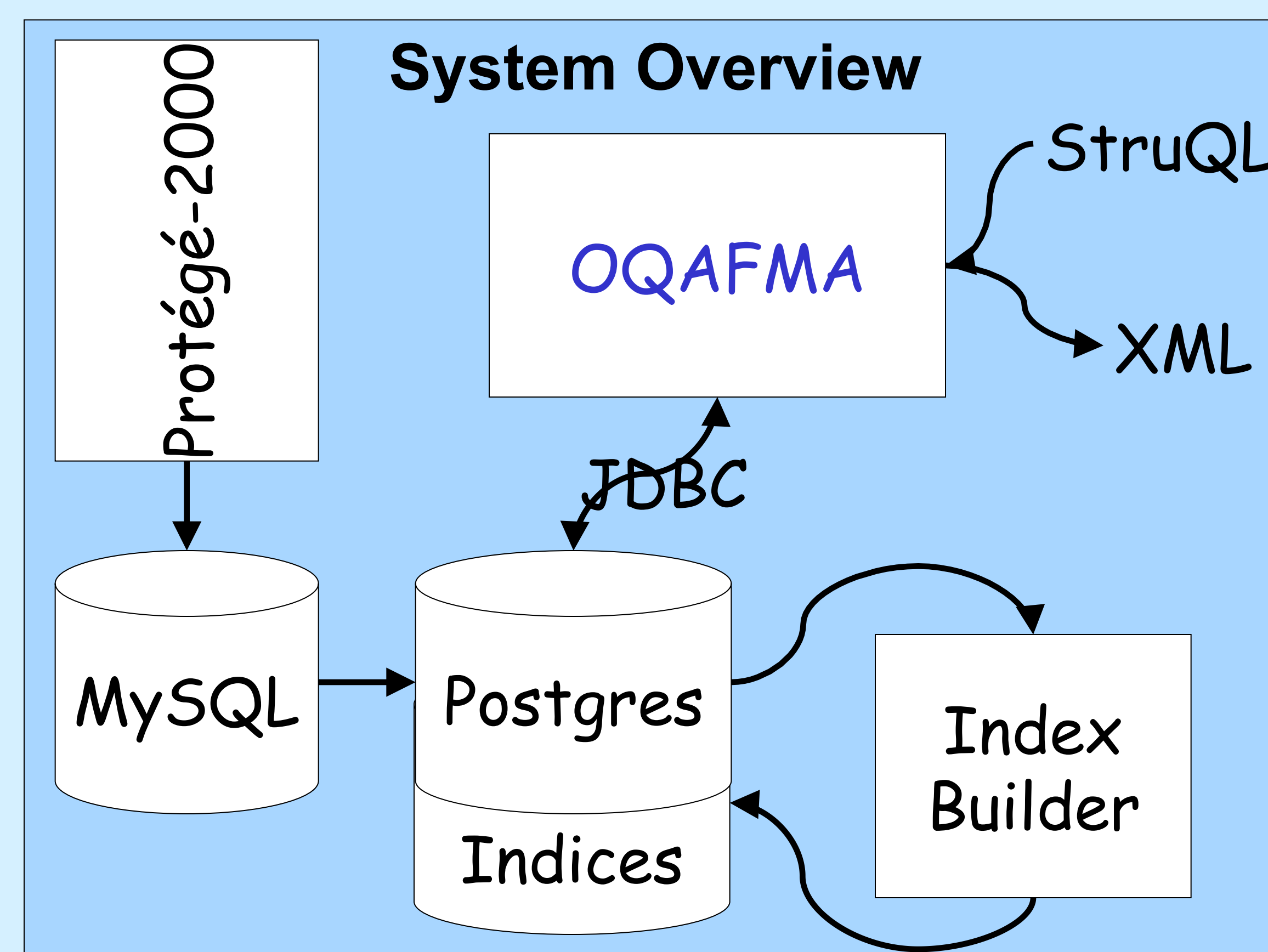
OQAFMA was developed to assure both flexibility and speed. It uses a subset of StruQL⁵ to provide nearly the same flexibility as the Protégé-2000 API. StruQL queries are posed by defining regular expressions over the relationships in the ontology (of which there are currently 122).

System Architecture

The figure below provides an overview of the system architecture. The FMA is currently stored in a MySQL database. The contents of this database are periodically transferred to a Postgres database to leverage the more powerful querying capabilities of Postgres. Once the data have been transferred, separate tables are constructed for each relationship in the ontology. This provides faster access to any given relationship.

In addition, the transitive closure of any relationship that binds an entity to another entity (as opposed to a value) is pre-computed. This is because closure operators are, most often, applied to a single relationship. For example, “find all subclasses of some class” is a closure operation over one relationship, namely the subclass relationship.

OQAFMA is implemented as a stand-alone server written using Java, JavaCUP and JLex. It accepts ASCII StruQL commands and returns XML.



Sample StruQL and XML Results

```
WHERE
X->":NAME"->"Thorax",
X->"generic part"*
      "contains"->Y,
Y->":NAME"->Contains
```

```
CREATE
```

```
TheThorax(Contains)
```

Note that one of the parts of the thorax is the thoracic cavity, which contains the lungs.

```
<results>
  <TheThorax>
    <Contains>Lung</Contains>
  </TheThorax>
  <TheThorax>
    <Contains>Upper lobe of right lung</Contains>
  </TheThorax>
  <TheThorax>
    <Contains>Right lung</Contains>
  </TheThorax>
  <TheThorax>
    <Contains>Upper lobe of left lung</Contains>
  </TheThorax>
  <TheThorax>
    <Contains>Left lung</Contains>
  </TheThorax>
</results>
```

Query Interface

Applications access OQAFMA using a declarative query language. This frees the application developer from needing to construct an access plan for each application (for example using the Protégé-2000 API). Instead, OQAFMA translates a StruQL query into a complex SQL query over the relationship tables. The database optimizer is responsible for identifying an efficient access plan.

Currently, the query interface supports a subset of StruQL. This subset includes optional relationships (?) and closure over single relationships (+, *). Multiple closures can be alternated (|) and finally concatenated (.). Since OQAFMA went online more than 12 months ago, no query of interest has been identified that cannot be expressed using this subset.

Additional applications are currently being written that use OQAFMA. In addition, the FMS has been rewritten using OQAFMA as its back end.

Sample Query and Result

The query (on the left) retrieves everything contained in some subpart of the Thorax. X is bound to the node whose name is Thorax. Y is bound to all nodes reachable from X via any number of part relationships and one contains relationship, and contains to those nodes' names. Due to incomplete population of the “contains” relationship, the results (on the right) are sparse.

Acknowledgements

This work was supported in part by NLM grants LM06316, 1T15LM007442-01, FMGrant.

References

- 1: Rosse C, Shapiro LG, Brinkley JF. The digital anatomist foundational model: principles for defining and structuring its concept domain. AMIA 1998: 820—824.
- 2: Stalder D, Brinkley JF. The digital anatomist foundational model server. O' Reilly Perl Conference 3.0 1999: 1—10.
- 3: <http://protege.stanford.edu>
- 4: Chaudri V, Farquhar A, Fikes R, Karp P, Rice J. Open Knowledge Base Connectivity 2.0.3. 1998 Apr 9.
- 5: Fernandez M, Florescu D, Levy A, Suciu D. A query language for a web-site management system. SIGMOD Record 1997 Sep; 26(3):4—11.