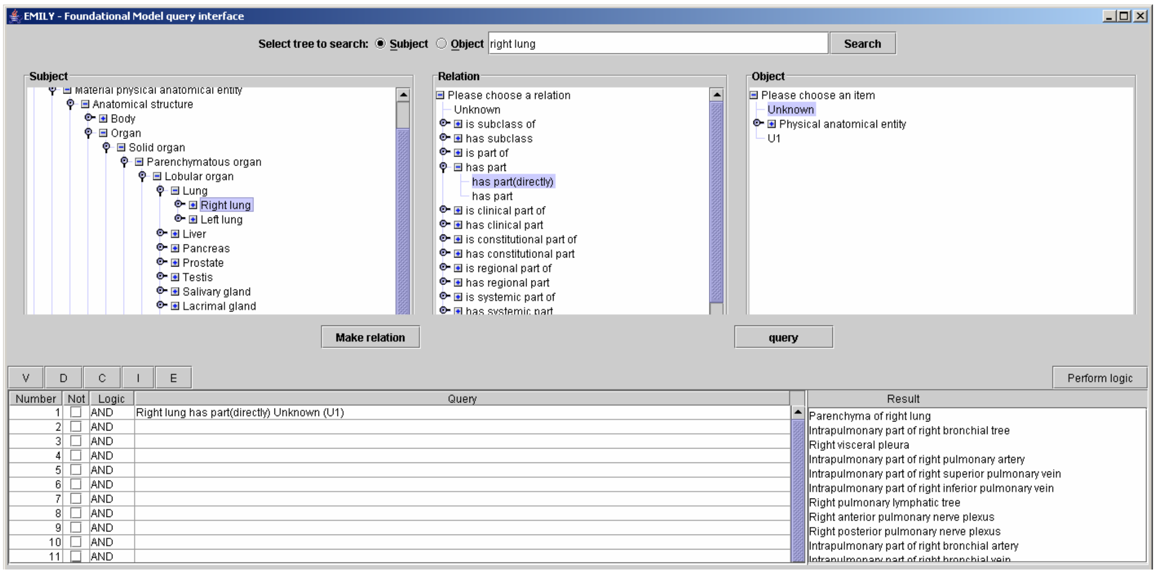
An Intuitive Graphical Query Interface for Protégé Knowledge Bases

Landon Todd Detwiler, Cornelius Rosse, MD, DSc, Linda Shapiro, PhD

Structural Informatics Group, Departments of Biological Structure, Medical Education and Biomedical Informatics, and Computer Science and Engineering
University of Washington, Seattle, WA 98195

Emily is a graphical query engine for Protégé knowledge bases that was developed by the Structural Informatics Group (SIG) at the University of Washington. Currently this application is adapted for a specific knowledge model, the Foundational Model of Anatomy (FMA) [1], but it could readily be generalized for use with other Protégé knowledge bases. In developing the *Emily* query interface, our intent was to provide a tool that was simple and intuitive to use, like the Queries tab provided with Protégé, but with improved information retrieval capabilities. Although some more advanced query mechanisms exist, currently they are too complicated for non-expert end users. The Algernon tab [2], for example, provides extensive Protégé query capabilities but requires users to learn a query scripting language. We sought to develop a query interface that was intuitive enough for end users to operate, with only minor instruction, yet was powerful enough to gather interesting information from a knowledge base that was not easily attained by browsing alone.


EMILY - Foundational Model query interface
Select tree to search: Subject Object
Search
Subject
Relation
Object
Make relation
query
V
D
C
I
E
Perform logic

Select tree to search: [✓ Subject] Object — right lung

Subject:
- Material physical anatomical entity
  - Anatomical structure
    - Body
    - Organ
      - Solid organ
        - Parenchymatous organ
          - Lobular organ
            - Lung
              - Right lung
              - Left lung
            - Liver
            - Pancreas
            - Prostate
            - Testis
            - Salivary gland
            - Lacrimal gland

Relation:
- Please choose a relation
  - Unknown
  - is subclass of
  - has subclass
  - is part of
  - has part
    - has part(directly)
    - has part
  - is clinical part of
  - has clinical part
  - is constitutional part of
  - has constitutional part
  - is regional part of
  - has regional part
  - is systemic part of
  - has systemic part

Object:
- Please choose an item
  - Unknown
  - Physical anatomical entity
  - U1

Number	Not	Logic	Query
1	☐	AND	Right lung has part(directly) Unknown (U1)
2	☐	AND	
3	☐	AND	
4	☐	AND	
5	☐	AND	
6	☐	AND	
7	☐	AND	
8	☐	AND	
9	☐	AND	
10	☐	AND	
11	☐	AND	

Result
Parenchyma of right lung
Intrapulmonary part of right bronchial tree
Right visceral pleura
Intrapulmonary part of right pulmonary artery
Intrapulmonary part of right superior pulmonary vein
Intrapulmonary part of right inferior pulmonary vein
Right pulmonary lymphatic tree
Right anterior pulmonary nerve plexus
Right posterior pulmonary nerve plexus
Intrapulmonary part of right bronchial artery
Intrapulmonary part of right bronchial vein

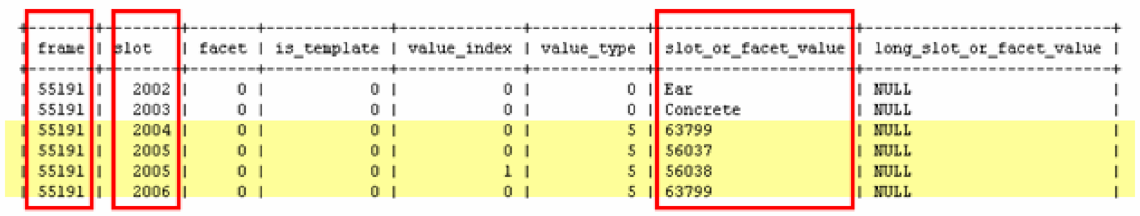
Figure 1: The *Emily* interface.

The *Emily* query tool, a standalone application built on top of the Protégé API, was constructed specifically for exploring the relationships that exist between knowledge base classes. The upper portion of the *Emily* interface is divided into 3 regions labeled Subject, Relation, and Object (*Figure 1*). Both the Subject and Object fields contain knowledge base classes arranged according to the superclass hierarchy. The center, Relation field contains a list of slots whose values are of type class[1]. The choices in each of these 3 fields are arranged into hierarchies and each field also provides an “Unknown” node. A search function is included to enable class location within the Subject

[1] Some slots of type instance are allowed for limited processing of reified relationships. For example, the Emily relationship “is adjacent to” actually maps to the FMA slot concatenation “adjacency”+“related object”, where “adjacency” can only contain values which are instances of the class “Anatomical coordinate adjacency” whose slot “related object” contains values of type class. The other information associated with Anatomical coordinate adjacencies (coordinate and laterality) are not accessible in the current implementation of Emily.

and Object hierarchies. This search field allows wild card as well as synonym searches[2]. To sufficiently describe a query, users must select a known value from at least 2 of the 3 fields. The remaining field can be specified by the user or he/she can select the Unknown entry. When choosing an entry from the relation field, users can either specify the direct form of a relationship, or its transitive closure, if appropriate. Once the query has been sufficiently described, a "query" button enables query processing.

The Subject, Relation, Object triple is a very intuitive form for most users as it mimics the structure of a simple English sentence (subject-verb-predicate). However, it is also a very appropriate match for the Protégé conceptualization of the represented knowledge, as a semantic graph whose class nodes form the pool of acceptable Subjects and Objects and whose directed edges describe the relationships between these nodes. Protégé's database backend schema is also constructed with the subject-verb-predicate form, with a few extra attributes included (see Figure 2).



```
+-------+-------+-------+-------------+-------------+------------+---------------------+--------------------------+
| frame | slot  | facet | is_template | value_index | value_type | slot_or_facet_value | long_slot_or_facet_value |
+-------+-------+-------+-------------+-------------+------------+---------------------+--------------------------+
| 55191 |  2002 |     0 |           0 |           0 |          0 | Ear                 | NULL                     |
| 55191 |  2003 |     0 |           0 |           0 |          0 | Concrete            | NULL                     |
| 55191 |  2004 |     0 |           0 |           0 |          5 | 63799               | NULL                     |
| 55191 |  2005 |     0 |           0 |           0 |          5 | 56037               | NULL                     |
| 55191 |  2005 |     0 |           0 |           1 |          5 | 56038               | NULL                     |
| 55191 |  2006 |     0 |           0 |           0 |          5 | 63799               | NULL                     |
```

Figure 2: Excerpt from FMA in Protégé database backend format. *Emily* is constructed to investigate the relationships between classes (value_type = 5). *Emily*'s Subject, Relation, Object fields roughly correspond with the frame, slot, and slot_or_facet_value database fields respectively (provided that facet = 0).

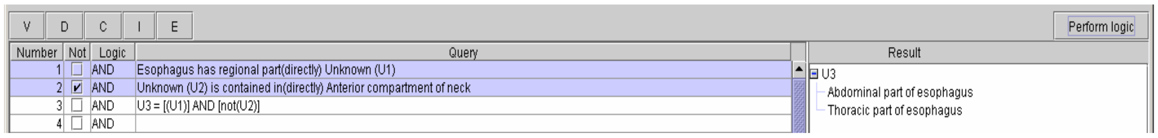
When a query is executed in which all 3 fields are specified, a Boolean value is returned. For example, in the FMA[3], the query "Heart has part Right atrium" would return the value "true". However, if either the Subject or Object is left as Unknown, then *Emily* returns a list of classes that could replace the Unknown in the query definition and would result in a "true" response. For example, the query "Right atrium has constitutional part (directly) Unknown" would return the set of values {Wall of right atrium, Cavity of right atrium, and Tricuspid valve} all of which are acceptable replacements for the Unknown. Note, that the similar query, "Right atrium has constitutional part Unknown" returns the transitive closure of the constitutional parts of the Right atrium (the parts previously mentioned, the constitutional parts of each of these, and so on). These are examples of basic *Emily* queries.

Each time an *Emily* query is executed, for which the result set contains one or more classes, a new node is automatically created for both the subject and object trees. These new node represents the result set. If one of these set nodes is double-clicked, it will produce a dialog box listing the members of the set. These set nodes can be used in subsequent queries just like any of the other class nodes. For example, suppose that a user wishes to know what entities are contained in some direct constitutional part of the Right atrium. The query "Right atrium has part (directly) Unknown" produces a 3 element result set, as mentioned above, which is assigned the name "U1" and is added to both the Subject and Object trees. A second query, "U1 contains Unknown" checks for containment relationships for each element in U1. The results are displayed as a tree with elements of U1 listed below the root node and their containments listed as children of these higher level nodes. In the FMA only spaces are allowed to have containment relationships, so the resulting tree has one child node below the root, "Cavity of right atrium", which itself has a child node "Blood in right atrium". The set of leaves from this query (in this case there is only one set member, "Blood in right atrium") becomes a new set U2 that is also added to the Subject and Object trees. This gives us one way of creating composite queries.

The bottom of the *Emily* interface shows a history of the queries executed, the variable names assigned to each result set, and a results display for each query. In addition, it provides us with another mechanism for creating composite queries, Boolean combinations of result sets. *Emily* allows us to combine results sets using conjunction, disjunction,

[2] Note, synonym searching is FMA specific functionality.
[3] All query examples used in this document are based on the Foundational Model of Anatomy knowledge base.

and negation operators. For example, if a user wishes to ask, “What parts of the esophagus are not contained in the anterior compartment of the neck?” One way to pose this query is the following: “Esophagus has part Unknown” AND NOT “Unknown is contained in Anterior compartment of neck” (see Figure 3).



Number	Not	Logic	Query
1	☐	AND	Esophagus has regional part(directly) Unknown (U1)
2	☑	AND	Unknown (U2) is contained in(directly) Anterior compartment of neck
3	☐	AND	U3 = [(U1)] AND [not(U2)]
4	☐	AND	

Result

U3
- Abdominal part of esophagus
- Thoracic part of esophagus

Figure 3: Boolean combinations of queries. Here two result sets were combined using Boolean operators. The “Logic” field allows user to select AND or OR, and negation can be achieve by selecting the “Not” field. The button “Perform logic” enables processing of the composite query.

So far we have mentioned what happens when either the Subject or Object fields are left Unknown, or values are supplied for all 3 fields. But, what happens when the relation field is chosen to be Unknown? This type of query requires special attention. The FMA is a highly connected network. Every class node is connected to every other class node by many paths. It is not practical to determine all possible paths in such an enormous network[4]. *Emily* returns the first path that it finds according to the following procedure, which assumes that the most direct connection is also the most desirable:

1. Check for direct connections. Is there a single edge (slot) connecting the Subject and Object entities in the knowledge base?
2. Check for transitive closure connections. Is there a path in the semantic graph, such that all edges are of the same type, that connects the Subject and Object entities?
3. Check for predetermined connection types. Is there a path in the semantic graph, such that the sequence of edges along the path matches a pattern pre-identified as significant? In this case, some sequences, such as the one identified by the regular expression part*contains[5] have been pre-encoded as significant. Currently these are hard-coded, but we envision them coming from a user editable configuration file.
4. Do a depth-limited breadth first search. If steps 1 through 3 fail to produce a more direct path, we resort to searching. Our current depth limit (of 4 edges) is chosen mainly for complexity reasons; however, in practice relationship chains longer than this are rarely desirable.

Emily provides a simple, intuitive interface for entering basic subject-verb-predicate type queries, transitive closure queries, and composite queries formed by reusing previous result sets and/or by performing Boolean operations on previous result sets. End users without prior experience should have little difficulty understanding the interface mechanisms, given minimal instruction. Restricting selection of query elements to those contained within the 3 component trees helps restrict the range of possible queries to those covered by the knowledge base domain (as opposed to more freeform interfaces like those found in many natural language query systems). Initial evaluation of the *Emily* tool suggests that it can effectively answer real world questions provided that such questions are restricted to the knowledge base domain and provided users can appropriately translate the question into *Emily* Subject, Relation, Object triplets [3].

References:

1. Rosse C, Mejino JLV. A reference ontology for bioinformatics: the Foundational Model of Anatomy. Journal of Bioinformatics. 2003;36(6):478-500.
2. For further information about Algernon see: http://algernon-j.sourceforge.net/
3. Shapiro LG, Chung E, Detwiler LT, Mejino JLV, Agoncillo AV, Brinkley JF, et al. A query interface for evaluating relationships in a large biomedical knowledge base, the Foundational Model of Anatomy. Submitted to JAMIA.

[4] The FMA knowledge base contains over 70,000 classes; a subset of over 185,000 frames. In the FMA there are 170 slots in use with more than 1.5 million slot values.

[5] part*contains indicates any number of part edges followed by a single contains edge.