

# **The Comparative Anatomy Information System for the Foundational Model of Anatomy and Mouse Anatomy Ontology**

**Tejinder Kaur Judge**

Undergraduate Researcher  
University of Wisconsin-Madison

**Kremena Diatchka**

Undergraduate Researcher  
Oregon State University

**Linda G. Shapiro**

Faculty Mentor

**Katarzyna Wilamowska**

Graduate Student Mentor

**Ravensara Travillian**

Graduate Student Mentor

Technical Report  
Department of Computer Science  
University of Washington – Seattle  
August 26, 2005

## **Abstract**

The Foundational Model of Anatomy (FMA) [5] is a hierarchical description of the human anatomy implemented in the Protégé 2000 knowledge representation. It can be navigated using Protégé or the Foundational Model Explorer (FME) or queried using the query engine *Emily* [2], [3]. An approach called The Structural Difference Method [4] has been proposed to correlate the anatomy of two species. In this paper, we will present the design of the Comparative Anatomy Information System (CAIS) that allows a user to issue queries determining the similarities and differences between two species based on the Structural Difference Method.

## **Introduction**

The increasing amount of anatomical and medical information emerging from animal modeling in comparative medicine and comparative genomics has created the need for an information system that evaluates, organizes and manages the data being collected. In addition, the interdisciplinary nature of medical research, as well as the increasing spectrum of users, has further created a need for such an information system. The system needs to be versatile enough to meet the needs of different levels of users, be they scientists, clinicians or students.

Since most contemporary research in medical informatics involves mice, there is a need for scientists to be able to easily find the similarities and differences between the mouse and human anatomy to ensure the capability of further expanding the results of their research to humans. We attempt to fill the need created by building CAIS. The objective of our work is to build an information system that allows users to query similarities and differences between two species. As current research in medical informatics involves mostly mice, we have decided to base our system on mouse and human anatomy. It is a starting point for the system which can be expanded to include more species.

## **Comparative Anatomy Information System (CAIS)**

Using the Structural Difference Method [4], each species is represented by an attributed graph. To compare two species, a mapping between them must be constructed and stored in a data structure. We have access to the human anatomy which is stored using the FMA [5], and the mouse anatomy which is stored in the mouse anatomy ontology (MAO) [6]. Both the FMA and MAO are implemented using the Protégé 2000 frame-based knowledge representation system. The knowledge base in CAIS is filled with information from the FMA and MAO. It is unique as it also includes mappings between the two species.

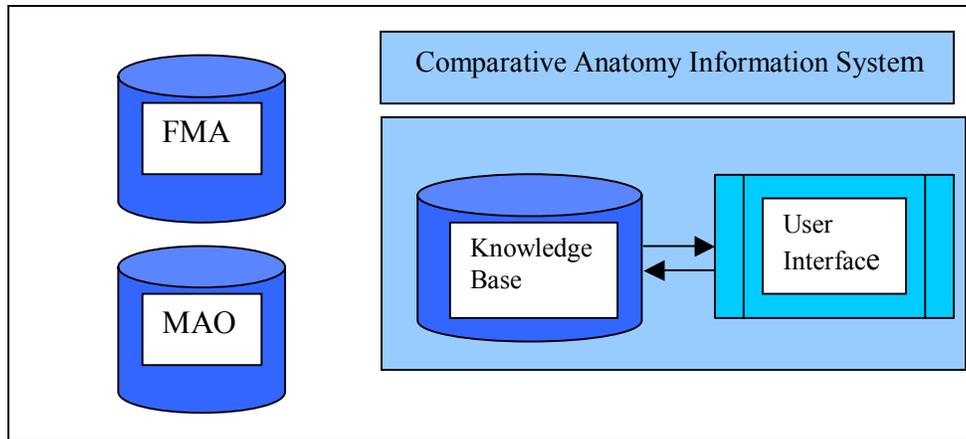


Figure 1: The knowledge base in CAIS is filled with information from the FMA and MAO. It is unique as it also includes mappings between the two species. The CAIS knowledge base is independent of the FMA and MAO. The user interface of CAIS accesses the knowledge base through methods calls using the Protégé 2000 library.

## The Knowledge Base

For our purposes, we have created a knowledge base using Protégé 2000 that stores information about the human anatomy, mouse anatomy and the mappings between them. The human anatomy data in our knowledge base is filled by using the information in the FMA [5]. The mouse anatomy is based on the MAO [6], while the mappings between them are currently being researched. Our knowledge base follows the structure of the FMA knowledge base which stores anatomical concepts as nodes and stores relationships between these nodes as edges connecting the nodes. Because Protégé is a frame-based model, every concept frame (i.e. relationships) is specified by a collection of slot values. Each node has attributes which are stored in slots and their relationships with other nodes are stored as slots as well. Due to time constraints, we currently only have created 22 slots. More slots can be added in the future when more types of relationships or attributes are found between the nodes.

For example, *Set of prostates (mouse)* and *Ventral prostate (mouse)* are two nodes in the knowledge base, and the relationship **has member** is a directed edge from *Set of prostates (mouse)* to *Ventral prostate (mouse)*. This is because set of mouse prostates has 5 members; ventral prostate, right dorsolateral prostate, left dorsolateral prostate, right coagulating gland and left coagulating gland. Since ventral prostate is one of the 5 members, there is an edge between *Set of prostates (mouse)* and *Ventral prostate (mouse)*. Among the slot values that are being used are **maps to**, **member of**, **part of**.

The nodes are stored as classes in Protégé. These classes are stored based on the is-a hierarchy. For example lobular organ is an organ and organ is an anatomical structure. As we are using the is-a hierarchy, some of the classes that appear are place holders, to describe a juncture in the hierarchy, in the example above, organ and anatomical structure. A separate, but equivalent hierarchy, is created by templates for each of these classes. The templates hierarchy is not required by Protégé, but is a design decision that

enables the inheritance of slots. Each class is an instance of its template and inherits the slots that are contained in the template. Example *Ventral prostate (mouse)* is an instance of the *Ventral prostate (mouse) template*.

## Mappings

A Mapping is a correspondence between a structure in the source (subject) species and a structure in the target (object) species (the subject is the species being compared to the object species). In a mapping direction, the subject is 'from' and the object is 'to'. For example, from human to mouse or from mouse to human. A mapping answers the question, "What is the structure in subject that corresponds to a structure in object?" For example "what is a structure in the human anatomy that corresponds to the *set of prostates (mouse)*?" The answer would be *prostate (human)*. This simply means that the human prostate is a structure in the human that maybe has *the same function*, *is in the same location* or *has structural similarities* to the set of mouse prostates. So we say *prostate (human) maps to set of prostates (mouse)*. These mappings are currently researched and are a crucial part of our information system. The mappings are the essential link between the human and mouse anatomy.

Thus the mapping which is stored in the slot *maps to* contains pointers in both directions between species i.e. the human can be the subject or object species and so can the mouse. Both directions are necessary to ensure that the result of the query "*prostate (human) is different from set of prostates (mouse)?*" and "*set of prostates (mouse) is different from prostate (human)?*" returns the same result.

## Implementation

After creating the knowledge base, we developed a Java application that serves as a Graphical User Interface (GUI) to the knowledge base. It allows a user to perform queries on the similarities and differences between the human and mouse anatomy. These queries are treated as method calls to the Protégé 2000 library, which performs the information retrieval. The system then translates the results into two forms, textual and tree representation, which are displayed for the user. Users can perform boolean and set queries. These queries can be performed on a single species (ex. human -> human) or on two species (ex. mouse -> human). The system has been developed to scale to larger numbers of species.

## Queries

### Boolean queries

Boolean queries can be of the form *is different?* and *is homologous?* The following is the semantics and results that are returned using these queries.

- *Species1.anatomical-entity1 is different? species2.anatomical entity2* returns TRUE if species1.anatomical-entity1 does not map to species2.anatomical-entity2, and FALSE if the two entities map to each other.

- `Species1.anatomical-entity1 is homologous? species2.anatomical entity2` returns FALSE if `species1.anatomical-entity1` does not map to `species2.anatomical-entity2`, and TRUE if the two entities map to each other. *is homologous?* is the inverse of *is different?*

## Set queries

The five set query operators are *differs from*, *similar to*, *shared*, *not shared* and *union*.

- `Species1.anatomical-entity1 differs from species2.anatomical entity2`  
Returns the difference between `anatomical-entity1` in `species1` and `anatomical-entity2` in `species2`. If they are isomorphic, it will return null.

Differences are determined by checking which slots `anatomical-entity1` has that `anatomical-entity2` does not have and vice versa. The slot values are also compared. Thus the different slots and slot values are displayed. These results are obtained with the default recursion level, level 0. If the recursion level is increased to 1, the results mentioned above are displayed. In addition, the parts of each entity are obtained and their slots and slots values are compared. The different slots and slots values are displayed.

- `Species1.anatomical-entity1 similar to species2.anatomical entity2`  
Returns the similarities between `anatomical-entity1` in `species1` and `anatomical-entity2` in `species2`. If they are different, it will return null.

Similarities are determined by the same method described above with the only difference being that same slot and slot values are the results being searched for.

In the implementation of the *differs from* and *similar to* queries, the first step is to check if the anatomical entities map to each other. Thus the query “how is the *ear (human)* similar to the *tail (mouse)*?” would return an empty result set as the human ear does not map to the mouse tail. On the other hand, for the query “how is *ear (human)* similar to *ear (mouse)*?”, the human ear maps to the mouse ear so it passes the first check and further comparisons are done.

- `Species1.anatomical-entity1 shared species2.anatomical entity2`  
returns the set of parts in `anatomical-entity1` and `anatomical-entity2` that map to each other.

The first step in this query is to obtain the parts of `anatomical-entity1` and parts of `anatomical-entity2`. Then we check if any of the parts of `anatomical-entity1` **map to** the parts of `anatomical-entity2` and vice versa. These parts are the shared parts.

- `Species1.anatomical-entity1 not shared species2.anatomical entity2`

returns the set of parts in anatomical-entity1 and anatomical-entity2 that do not map to each other.

This query is performed as mentioned in *shared* with the only difference being that we are checking check if any of the parts of anatomical-entity1 do not **map to** the parts of anatomical-entity2 and vice versa. These parts are the not shared parts.

- Species1.anatomical-entity1 *union* species2.anatomical entity2  
returns the set of all shared and not shared mappings of anatomical-entity1 in species1 and anatomical-entity2 in species2.

In this query, we return the names of the classes that anatomical-entity1 **map to** and the classes that anatomical-entity2 **map to**.

For *shared*, *not shared* and *union*, it is not necessary for the subject entity and object entity to map to each other.

### **Unknown entity queries**

Unknown queries are used if for example a user wants to know “what is a structure in the human anatomy that corresponds to the *set of prostates (mouse)*?” The answer would be *prostate (human)*. This is because *prostate (human)* is the value in the **maps to** slot of *set of prostates (mouse)*. When selecting a subject entity or object entity, the user has the option of selecting *unknown* for either one of the entities. A query that involves *unknown* as the subject and a valid entity from the hierarchy as an object returns the value(s) in the valid entity’s **maps to** slot. If the same valid entity was chosen as the subject and *unknown* as the object, the corresponding result will be obtained. Example “what is a structure in the mouse anatomy that corresponds to prostate (human)?” The answer would be *set of prostates (mouse)*. A query involving two *unknowns* returns an error message. For a query involving an *unknown*, the query type chosen does not make any difference.

## The graphical user interface

The graphical user interface that we have developed is shown below.

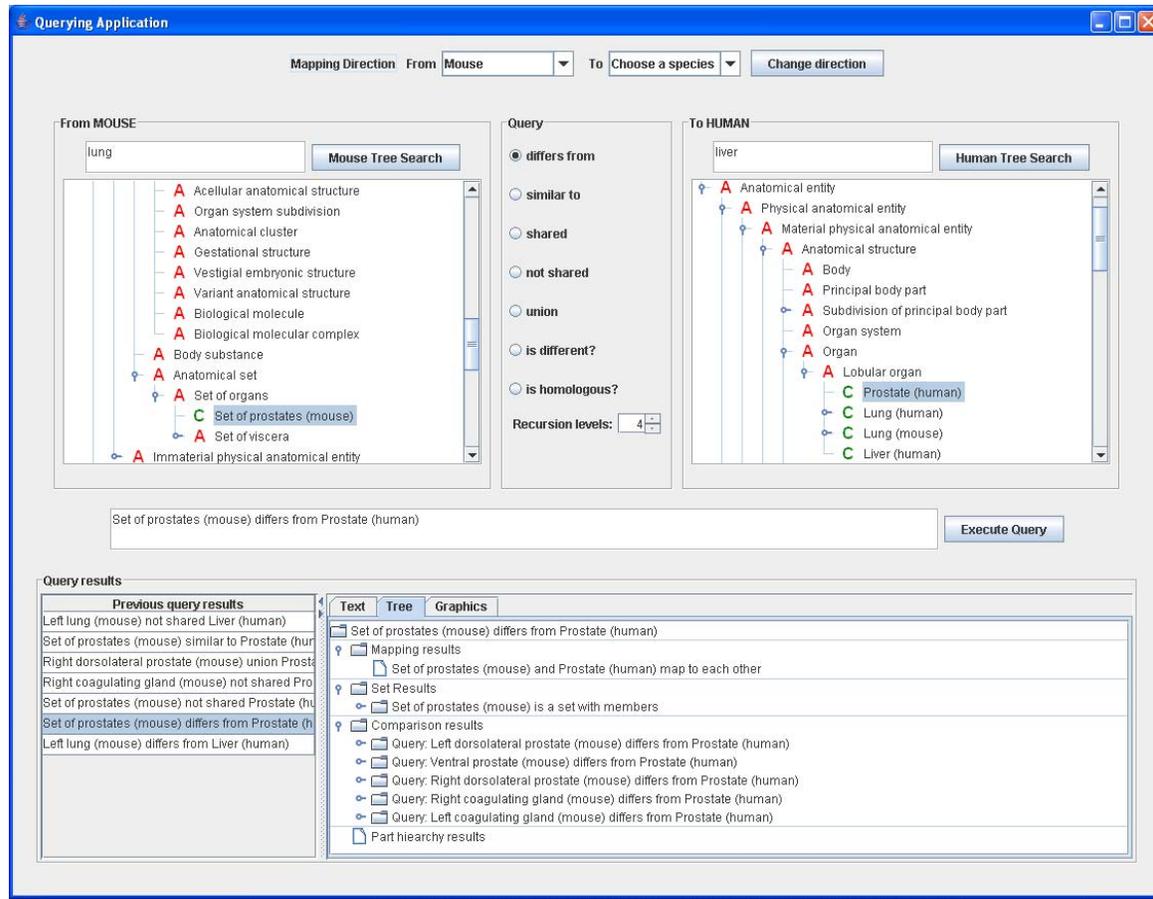


Figure 2: the graphical user interface after processing the queries 'Set of prostates (mouse) differs from prostate (human)'. The results panel shows the tree display.

### Mapping Direction

The Mapping Direction drop down box, allows the user to select one species as the subject (from) and one species as the object (to). It is set by default to show mouse as the subject and human as the object. A user has the option to make a single species (human -> human, mouse -> mouse) query or a two species (human -> mouse, mouse -> human) query. Although currently the system supports only the human and the mouse, it has been designed to scale to more species.

## Search Box

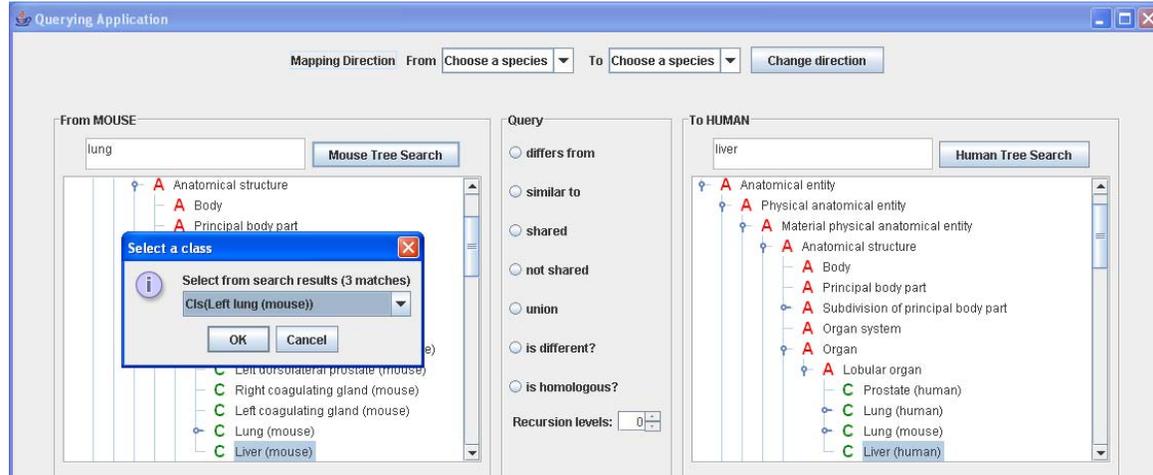


Figure 3: The mouse tree shows a pop up box displaying the three matches for the search 'lung'. The human tree show the class 'Liver (human)' selected after it was found using the search box.

After selecting the mapping direction, the user can search for an entity by using the search box or navigate through the hierarchy to find the anatomical entity he or she desires. The search box allows exact name searches as well as wildcard searches. If a user types lung in the mouse anatomy hierarchy, a pop up box will appear with three matches, *lung (mouse)*, *right lung (mouse)* and *left lung (mouse)*. He or she will be able to choose from the selection. If there is only one match (ex. liver in the human tree) the class will be found and selected. If there are no matches, a pop up box saying 'No match was found' will be displayed.

## Tree Navigation

The + sign by the structure name indicates that a hierarchy can be expanded. We expand the hierarchy along the is-a structure. For example organ is part of anatomical structure. As we are using the is-a hierarchy, some of the classes that appear are place holders to describe a juncture in the hierarchy. These classes are known as abstract classes and cannot be used as part of a query. The user is not able to select an abstract class. He or she can further differentiate between abstract classes by the red icon bearing the symbol 'A' as compared to classes that can be selected (concrete classes) that have a green icon with the letter 'C'.

## Query Type

The Query radio buttons allow the user to select a query relationship that will use the specified subject entity and object entity as arguments. For queries that support recursion, the user can determine the level of recursion required. The default value is set to 0.

## **Query Information and Query Execution Button**

The query information box is used to provide instant feedback to the type of query the user has chosen to execute. It shows the subject entity, query type and object entity. The query execute button executes the query.

## **Query Results**

The Query Results panel contains the Results Display and a Previous Queries Pane.

The Results Display panel has three tabs: text, tree and graphics. The text tab displays results textually while the tree tab displays the results in a tree data structure. Levels of recursion are seen more clearly using the tree display as each level is a node in the tree. Displaying the results in a graphical form is still under implementation.

Previous Queries are saved and the results can be seen by clicking on them. Clicking on a previous query, executes the query again displaying the result.

## **Future work**

The CAIS described in this paper is the system developed during the duration of the author's research stay at the University of Washington-Seattle. The knowledge base that stores mappings of the human and mouse anatomy is still under construction. For our purposes we have one complete class in each species: The human prostate and set of mouse prostates with its five members, ventral prostate, right coagulating gland, left coagulating gland, right dorsolateral prostate and left dorsolateral prostate. Since the knowledge base is not complete, we have not been able to test our system using real data. For the development of our interface, we used mock test classes and at times the prostate class. We have also not been able to test the interface on a real set of possible users.

## **Conclusion**

Contemporary research mainly in bioinformatics, genomics, proteomics and other fields, have created a need to extrapolate information from one species to another. The amount of anatomical and medical information emerging from animal modeling is increasing the need for a system that evaluates, organizes and manages that information. In addition, the expanding spectrum of users for such an information system; from students to scientists or clinicians, emphasizes the importance of such an information system to accommodate needs of different users.

In this paper, we have described the Comparative Anatomy Information System (CAIS) to meet this need. Our system uses the FMA [5] as a framework and The Structural Difference Method [4]. It is able to answer queries regarding cross-species similarities or differences, making it a useful tool in both medical informatics and comparative anatomy.

## **Acknowledgement**

The author's research stay at the University of Washington-Seattle was sponsored by the Distributed Mentor Program (DMP), part of the Committee on The Status of Women in Computing Research (CRA-W).

## **References**

- [1] Travillian RS, Gennari JH, Shapiro LG. Of Mice and Men: Design of a Comparative Anatomy Information System. *Proceedings of the AMIA Annual Fall Symposium*, November, Washington DC, 2005.
- [2] Detwiler LT, Chung E, Li A, Mejino Jr. JLV, Agoncillo AV, Brinkley JF, Rosse C, Shapiro LG. A Relation-Centric Query Engine for the Foundational Model of Anatomy. *Proceedings, MedInfo 2004*, pages 341-345, San Francisco, CA.
- [3] Shapiro LG, Chung E, Detwiler LT, Mejino Jr. JLV, Agoncillo AV, Brinkley JF, Rosse C. Processes and Problems in the Formative Evaluation of an Interface to the Foundational Model of Anatomy Knowledge Base. *Journal of the American Medical Informatics Association*, volume 12, pages 35-46, Jan/Feb 2005.
- [4] Travillian RS, Rosse C, Shapiro, LG. An Approach to the Anatomical Correlation of Species through the Foundational Model of Anatomy. *Proceedings, American Medical Informatics Association Fall Symposium*, pages 669-673, 2003.
- [5] Rosse C, Mejino Jr. JLV. A reference ontology for biomedical informatics: the foundational model of anatomy. *Journal of Biomedical Informatics*, volume 36, pages 478-500, 2003.
- [6] The Jackson Laboratory (JAX) Mouse Genome Informatics. Available at <http://www.informatics.jax.org/>.