

# **A Set and Pattern Centric Query Engine to Explore the Ontology of Craniofacial Development and Malformation**

**Author: Emilia Holbik**

**Contributors: Linda G. Shapiro, Melissa D. Clarkson, Landon T. Detwiler, José L.V. Mejino Jr., James F. Brinkley**

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

## **Abstract**

*The Ontology of Craniofacial Development and Malformation (OCDM) is a computable representation of knowledge describing the anatomy of mouse and human heads [1]. The Cranio-Dynamic query interface is a tool with a graphical user interface for querying the OCDM. This project builds on approaches used within the Emily tool which queried the Foundational Model of Anatomy (FMA), an ontology for the domain of human anatomy [2]. The newly constructed web-based tool creates set-centric navigation capabilities on the current OCDM framework, introduces navigation capabilities based on intuitive groups of properties (pattern-centric navigation), and allows users to create customized patterns.*

## **Introduction**

Cranio-Dynamic is a tool designed to allow people to retrieve knowledge from the Ontology of Craniofacial Development and Malformation (OCDM) using a graphical interface. This tool relies on the Query Integrator web service to execute queries and return results. Since the OCDM has a database of over 23,000 anatomical structures, an efficient Query Integrator tool, and an intuitive web-accessible page, Cranio-Dynamic has the potential to make the OCDM more accessible to craniofacial specialists, students, and health care providers.

The objective of this paper is to describe Cranio-Dynamic and its underlying framework, illustrate the user-intuitive nature of the query engine, and introduce the concepts of pattern matching and configuration for more concise navigation. Before discussing these objectives in great detail, we will describe the OCDM and its method of navigation.

## **Ontology of Craniofacial Development and Malformation**

Funded by the National Institute of Dental and Craniofacial Research (NIDCR), the FaceBase Consortium (<https://www.facebase.org>) focuses on developing common software tools and craniofacial research data that can be shared among scientists around the world [3]. FaceBase also supports the construction of ontologies, which represent knowledge about a particular domain in the form of terms

and relationships [4]. Ontologies are useful because they are both human-readable and computer-parsable. The OCDM, a part of a spoke project of the FaceBase Consortium, currently consists of five craniofacial sub-ontologies: canonical human ontology (including developmental anatomy), canonical mouse ontology (including developmental anatomy), abnormal human ontology, abnormal mouse ontology, and human-mouse mapping ontology.

The OCDM is authored using the Protégé frames interface. The frames version of the ontology is converted to OWL and the statements are stored in a triple store for the purpose of efficiency. Then, the Query Integrator (QI) [5] uses queries written in SPARQL to query the triple store, and it sends back RDF formatted as XML. The OCDM connects various terms across each of the five sub-ontologies using statements in the <Subject> <Property> <Object> triplet format, such as: <Right nasal bone> <bounded by> <Surface of right nasal bone> and <Face (Mus musculus)> <constitutional part> <Left eyeball (Mus musculus)>. Cranio-Dynamic was designed to explore groups of terms related by single or multiple properties.

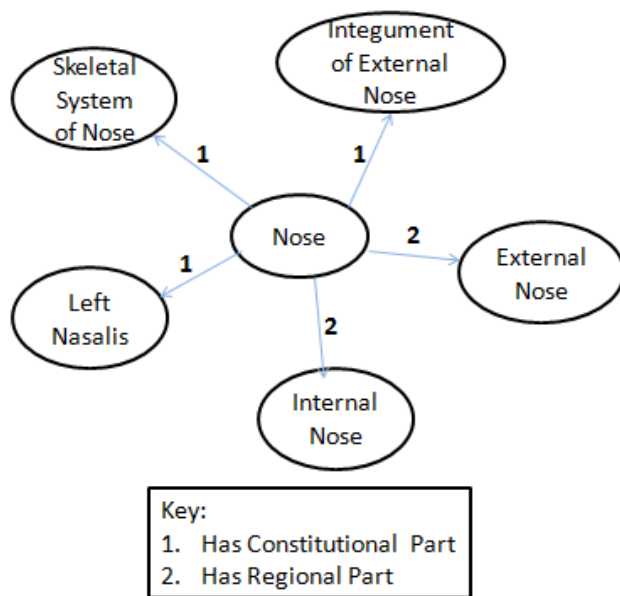


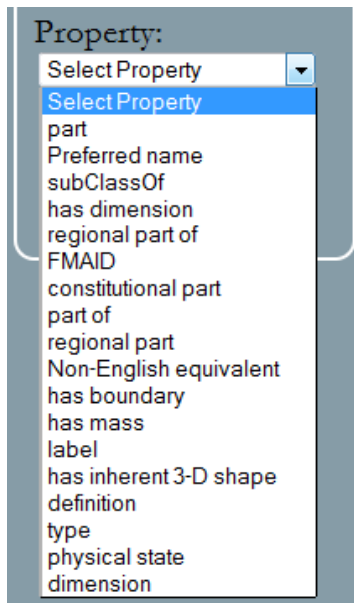
Figure 1: Nose is the subject, the directed lines are the properties, and the remaining bubbles are the objects.

## Navigating the OCDM

Since the information in the OCDM is communicated to Cranio-Dynamic using RDF, its structure can be visualized graphically in the triplet format. The subject is the starting point of the query and the objects are the destinations which are connected to the subjects via the property. *Figure 1* depicts a directed graph visualization of the properties and objects of *Nose*. Since the OCDM contains myriad properties that connect its subjects and objects, the efficiency of the Query Integrator's ability to convert information from the database into RDF and transmit it through the internet critically determines the response time of the Cranio-Dynamic user interface.

Querying the OCDM allows the user to ask questions using a pre-formatted question template and custom content. Cranio-Dynamic was designed with the goal of helping the user understand the format of the query question template as well as help the user insert subjects by browsing content that already exists. Furthermore, Cranio-Dynamic's dynamic system narrows possible queries to only those which will return valid results, as shown in *Figure 2*.

Figure 2: A dynamic list of properties.



## Cranio-Dynamic

Cranio-Dynamic is described as set-centric because it is most suitable for helping the user find a set (or group of objects) related to a chosen subject. Moreover, Cranio-Dynamic is described as pattern-centric because it uses patterns (or groups of properties) to simplify complex navigation through the OCDM. The following sections of this paper will explain Cranio-Dynamic’s framework and process of interaction with the OCDM database, an overview of Cranio-Dynamic’s GUI, and the various formats of queries that Cranio-Dynamic handles.

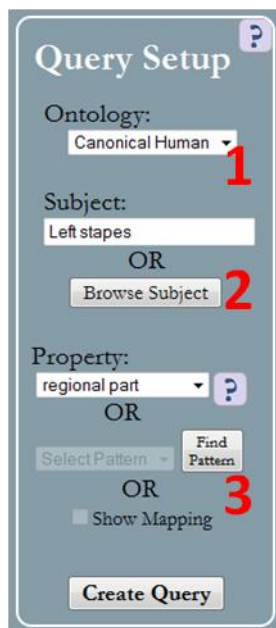
### Framework

Cranio-Dynamic’s graphical user interface is built using HTML 5, CSS, and JavaScript (including the Prototype) libraries. When the user creates and issues a query, JavaScript code creates a query URL which is then transmitted to the Query Integrator using AJAX [6]. The URL consists of

the following components: a host computer, path to the Query Integrator, query ID so the QI can identify a SPARQL query in its database, and the subject argument. Once the URL is sent, the Query Integrator retrieves the correct results from the OCDM OWL database in RDF using the SPARQL query language and sends the results back through the AJAX call to the JavaScript code. Then, the JavaScript code parses the RDF results in order to display them to the user in a more intuitive manner.

## Overview of Cranio-Dynamic GUI

Figure 3: A sequential guide to help the user set up a query



### a. Query Setup Panel

The Query Setup panel is located on the left side of the Cranio-Dynamic interface. First, the user must select an ontology from the drop down box, which lists four possible ontologies: Canonical Human, Canonical Mouse, Abnormal Human, and Abnormal Mouse. Second, the user may either type a class name as a subject or may browse for a subject using a part hierarchy, a feature which is currently only available in the Canonical Human and Canonical Mouse, as the others have no part hierarchies. Once a subject has been chosen, Cranio-Dynamic will dynamically list the properties available to the subject and will enable the “Show Mapping” checkbox if a mapping exists. If a user would like to query using a pattern instead, the user must click the “Find Pattern” button, causing Cranio-Dynamic to dynamically list the patterns available to the subject. Third, the user must choose a single property, a pattern, or a mapping prior to pressing the “Create Query” button. *Figure 3* shows the steps in query creation.

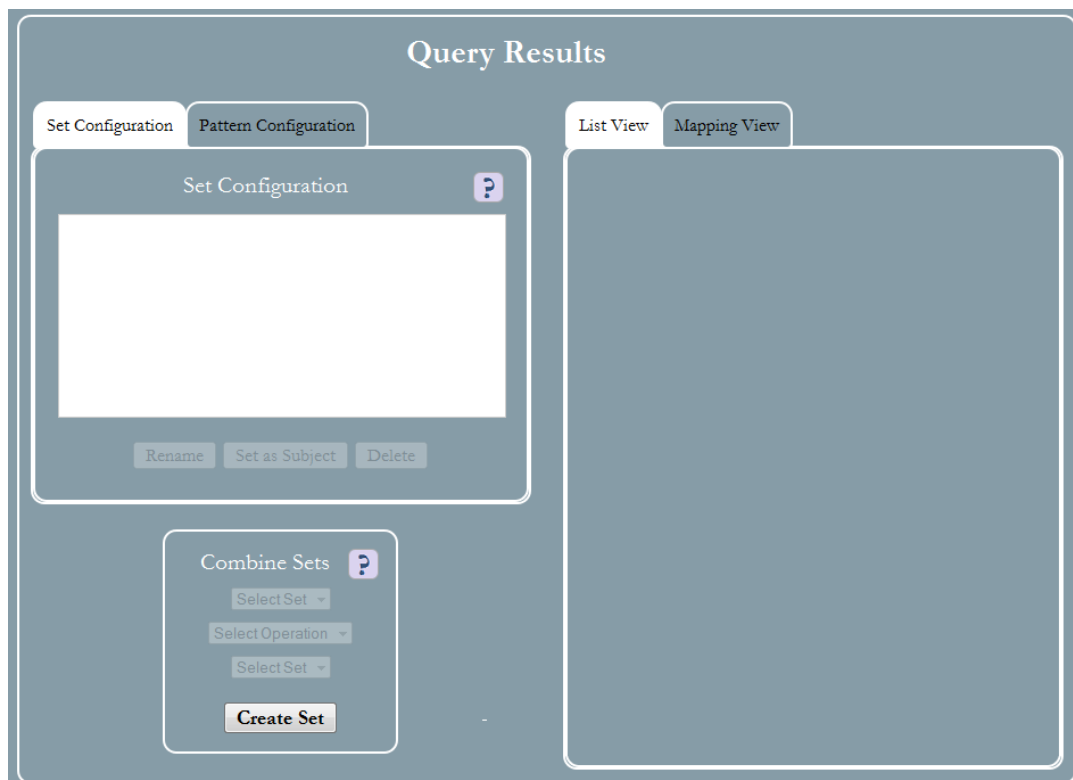
## b. Query Results Panel

Once the user has selected “Create Query”, the left and right columns of the Query Results panel will display results. The right panel will show a list of query results in the List View and various properties in the mapping ontology in the Mapping View (if the user checked the “Show Mapping” checkbox). The left panel will show the set created by the query in the Set Configuration panel and the list of possible patterns in the Pattern Configuration panel.

At the bottom of the Set Configuration panel in the Combine Sets panel, a user may customize a set. Using the drop down boxes in this panel, a user may merge two sets using any of the three following operations: AND, OR, and BUT NOT. The AND operation creates a new set by only adding elements that are present in both sets, the OR operation creates a new by merging two sets, and the BUT NOT operation creates a new set by adding elements that are present in the first but not second set.

Similarly to the Combine Sets panel, the Combine Properties panel is located at the bottom of the Pattern Configuration panel. When combining properties, the user has the opportunity to select a property which the query engine has encountered during this session of use. Furthermore, if the user selects the Repeat checkbox, the Kleene star will be applied to the property, allowing the property to be repeated 0 or more times in a query. If the Repeat checkbox is not checked, the property will be matched exactly once. *Figure 4* depicts the entire Query Results panel, the Set and Pattern Configuration panels, and the List and Mapping Views.

Figure 4: Displays an initial version of the Cranio-Dynamic GUI before the user has begun interacting with it.



Cranio-Dynamic also includes other useful features, many of which are similar to the features in the Emily [2] [6] system and many of which are new. Some of the similar features include the following: the ability to type in or browse for an anatomical term, a chronological history of all queries issued in the current session on the page, and clicking on saved sets in order to view previous results. New features include the following: help options in the top right corners of various elements of the GUI, a dynamically filtered list of properties pertaining to each subject, the ability to rename a set, and the Pattern Configuration tab.

## Query Variations

Using the Cranio-Dynamic graphical user interface, users can construct a basic query in the format of a triplet: <Subject> <Property> <Object>. The object is always the unknown entity in the query. The following subsections describe the query variations available to the user when selecting each parameter in the triplet.

### 1. Subject Variations

The subject denotes the starting point of the query and is the class (or set of classes) to which the property will be applied in order to retrieve results.

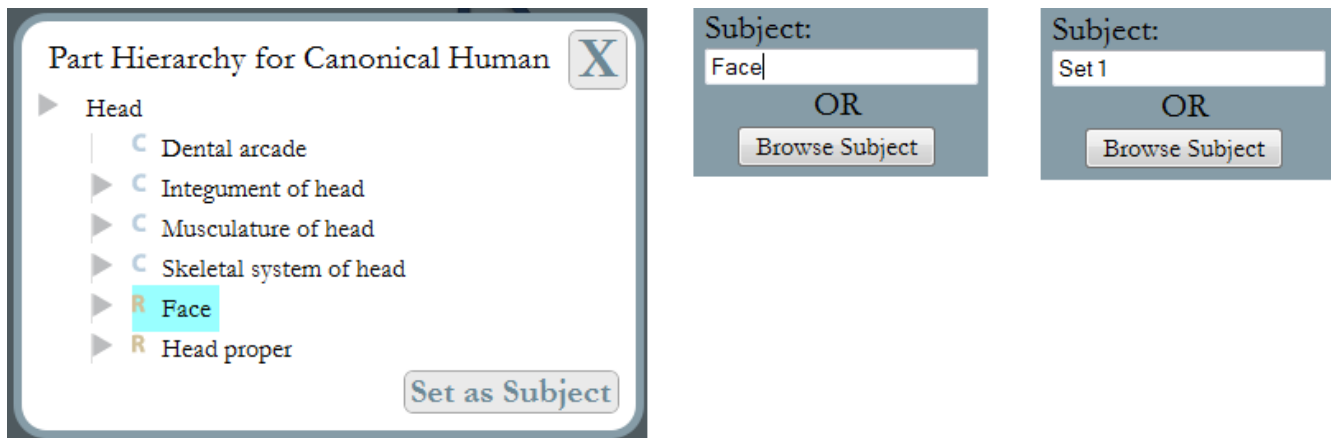
#### *a. Subject as Single Term*

Cranio-Dynamic allows the user to enter a single class from which to begin searching for results. The subject may come from any class in the OCDM, including those which fall under the human or mouse, canonical or abnormal categories. For example, *Right malleus* is an example of a subject with a rich list of properties. Some of these properties include: *articulates with*, *derives from*, *part of*, *subclass of*, *etc.*

#### *b. Subject as Set of Terms*

Once a user has created at least one query and retrieved a set of objects as a result, the user may decide to navigate more deeply into the OCDM by using the entire set of objects from a previous query as the subject of the current query. This is often referred to as “querying over a set.” When a user queries over a set, the property is queried for every member and every object is returned. For example, Set 1 contains the elements *Head proper* and *Head proper (Mus musculus)*. Furthermore, using Cranio-Dynamic, the user asks the OCDM <Set 1> <subclass of> <Unknown>. Then, Cranio-Dynamic breaks this set query into two parts: <Head proper> <subclass of> <Unknown> and <Head proper (Mus musculus)> <subclass of> <Unknown>, and retrieves results from both queries, which are then unified. Therefore, the final result is <Set 1> <subclass of> <Set 2>, where Set 2 consists of *Subdivision of head* and *Subdivision of head (Mus musculus)*.

Figure 5: Shows the three types of ways to insert a subject: by choosing a subject with the "Browse Subject" button, typing a class name, or typing a set name



## 2. Property Variations

The property describes the directed edge between the subject node and an object node. It identifies the property or set of properties which link a subject and an object.

### a. Singular Property

A singular property can be used in order to identify how a subject and object are related. If deciding to query using a single property, the user must choose from a dynamically prepared list of predefined properties which are associated with the subject. The property options listed by Cranio-Dynamic will yield valid, non null results since properties that are not present in the OCDM for that subject will not be shown. Depending on which property was chosen, Cranio-Dynamic will return other classes, Boolean values, or instances. For example, `<Scalp> <part of> <Head proper>` returns a class, `<External nose (Mus musculus)> <reviewed by expert> <false>` returns a Boolean, and `<Basicranium> <preferred name> <KB INSTANCE 66582>` returns an instance. Only subjects produce a list of properties, but Booleans and instances do not.

### b. A Group of Properties (Pattern)

A pattern is defined as a sequence of properties, or a set of directed edges to follow, beginning from the subject. The purpose of using the default patterns and creating custom patterns is to match how users think about anatomy. The user will not be able to view the classes through which the query traversed in order to retrieve the final set of object results. This decision was made based on the assumption that intermediate classes would be unnecessary and provide a potentially confusing amount of information. Cranio-Dynamic provides important and common OCDM-related default patterns to enhance the user's navigation capabilities and allows the user to create custom patterns from a list of previously-seen properties.

When querying using a pattern, the first property is applied to the first subject. After the first subject and property pair retrieves the relevant objects, those objects then become the subjects for the next property in the pattern, and so on. Therefore, each pattern has a sequential list of properties, where a single

**Figure 6: Displays the default patterns and shows the creation of Pattern 5, a custom pattern**

The screenshot shows a software interface with two tabs: "Set Configuration" and "Pattern Configuration". The "Pattern Configuration" tab is active, displaying a list of default patterns and a section for creating a custom pattern.

**Pattern Configuration**

Default 1: Given an anatomical structure, find any points associated with it (bounded by\regional part\*\bounded by\bounded by)

Default 2: Given an anatomical point, find the structure associated with it (bounds\bounds\regional part of\*\bounds)

Default 3: Walk down branched tree structure (regional part\regional part\*)

Default 4: Walk up branched tree structure (regional part of\*\regional part of)

**Pattern 5 (constitutional part\*\subClassOf)**

Rename Delete

**Combine Properties**

constitutional part ☒ Repeat -

subClassOf ☐ Repeat -

+

Create Pattern

property can be applied 0 or more times if the user

checks the “Repeat” checkbox. The following example of a pattern finds the points associated with an anatomical structure: *bounded by\regional part\*\bounded by\bounded by*, where each property is separated by a backslash. *Figure 6* shows the default patterns and the creation of a custom pattern.

### 3. Result Object Variations

As soon as the user creates a query, the results of that query are available in various formats which allow the user to set the result of a previous query as the subject of the current query. Prior to the creation of a query, the user must specify whether or not to retrieve mappings between species; however, a set of terms will be created regardless of user choice.

#### a. Result Object as Mapping

The subject in the ontology selected will be mapped to another subject in the corresponding ontology on the basis of evolutionary homology. The breakdown of ontology mappings is unidirectional:

*Canonical Human* → *Canonical Mouse*

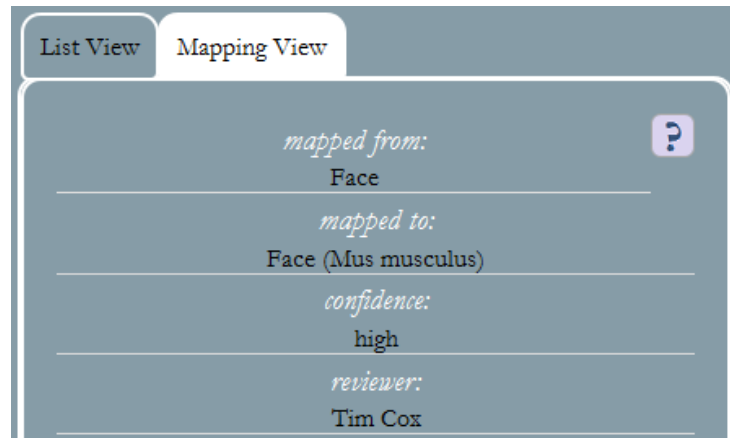
*Canonical Mouse* → *Canonical Human*

*Malformations of Human* → *Malformations of Mouse*

*Malformations of Mouse* → *Malformations of Human*

As mappings in the OCDM are currently structured, they will either be one-to-one mappings or one-to-null mappings. For example, Snout (Mus musculus) in the Canonical Mouse ontology does not map to any term in the Canonical Human ontology because humans do not have a snout. However, Face in the Canonical Human ontology and Face (Mus musculus) in the Canonical Mouse ontology map to one another, since humans and mice have faces.

Figure 7: Shows the results of the following mapping: <Face> <maps to> <Face (Mus musculus)>



### ***b. Result Object as a Set of Terms***

When a query is created, any results are returned in the form of a single set, (a set is similar to a “box” which holds zero or more result objects). The cardinality of the set is equal to the number of objects.

The creation of a set is motivated by the need to use results of previous queries as the subjects in current queries. A set becomes particularly useful when two or more result objects are necessary as subjects for subsequent queries. For example, if the user needs to find the regional parts of the regional parts of Neurocranium (Mus musculus), the user may query first to find the regional parts of Neurocranium (Mus musculus) and then make Set 1, the result of the query, also the subject of the next query and find its regional parts.

Even though mappings yield only zero or one objects and do not rely on the selection of a property, a set is still created to contain the result of the mapping and can be used in a subsequent query.

### **Query Examples**

The basic form of the <Subject, Property, Object (Unknown)> allows for the following different types of queries:

1. Mouth *has constitutional part* Unknown
2. Mouth “*Show Mapping*” *box checked* Unknown
3. Set 1 *has constitutional part* Unknown
4. Mouth *has Pattern 1* Unknown

### **Use of Patterns as Properties**

The purpose of using patterns is to hide the ontology’s complexity from a new user and to save common groups of properties for later navigation use. As soon as the query engine launches, multiple default example patterns are available for the user. For example, the user may traverse from an anatomical structure to its points or may walk branched tree structures using the default examples.



The following is a concrete example of traversing from an anatomical structure to any of its anatomical points:

```
<Right nasal bone> <bounded by> <Surface of right nasal bone>

    <has regional part*> <External surface of right nasal bone>

        <Internal surface of right nasal bone>

    <bounded by> <Superior border of right nasal bone>

        <Inferior border of right nasal bone>

        <Lateral border of right nasal bone>

        <Medial border of right nasal bone>

        <Internal anatomical border of right nasal bone>

    <bounded by> <Superolateral point of right nasal bone>

        <Nasion>

        <Rhinion>

        <Inferolateral point of right nasal bone>
```

## Reliability of Cranio-Dynamic

Without formal testing and examination but only through local and thorough tests of myriad examples (a few of which are described throughout this paper), Cranio-Dynamic returns the correct results. Furthermore, Cranio-Dynamic delivers results in an efficient manner, retrieving results in as little as 2 milliseconds to 10,000 milliseconds for complex queries.

## Discussion, Future Work, and Conclusion

This tool has been developed to explore how sets and patterns can be used in querying the OCDM. As seen from the discussion topics in this paper and from its results, Cranio-Dynamic is a promising tool to help students, anatomists, and public health professionals navigate the OCDM. As a part of future work, Cranio-Dynamic must undergo the following user tests:

- *Content evaluation:* This evaluation would test the fidelity with which Cranio-Dynamic returns query results.
- *User interaction evaluation:* This would assess whether or not Cranio-Dynamic is usable by the intended audiences.

However, throughout development, the creation of this tool has helped to discover future research directions. The following are suggestions for the continued improvement of the Cranio-Dynamic query engine:

- *Application of regular expressions to properties in a pattern:* Instead of only allowing a user to apply a Kleene star to a single property (which applies a property 0 or more times), allow the user to apply a plus (+) and question mark (?) operator. The plus operator would match a certain property 1 or more times and the question mark operator would match the property 0 or 1 times exactly. These operators have the same functionality as Linux regular expressions.
- *Pattern operations:* When creating a pattern, Cranio-Dynamic currently only allows properties to be applied to a subject sequentially (a.k.a. the resulting objects of a previous query become subjects of a subsequent query, and so on). We will name this sequential algorithm “THEN” operation. This functionality can be expanded to allow the resulting objects from multiple properties to be combined using the “AND”, “OR”, and “BUT NOT” operations. Therefore, the user will have a choice of four operations to apply between each property in the pattern.
- *Translation of patterns into SPARQL queries:* In order for queries to be usable outside of a single web session on Cranio-Dynamic, the queries would need to be translated from JavaScript to the SPARQL query language. This would help users overcome the hurdle of writing SPARQL code in order to extract certain information from the OCDM.
- *Browsing class hierarchies for subjects in addition to part hierarchies.* Users can be given a choice of hierarchy to browse.

## Acknowledgements

Research on the OCDM is funded by the National Institute of Dental and Craniofacial Research under Grant No. U01-DE020050. Furthermore, Emilia Holbik’s work is funded by the Computer Research Association’s Committee on the Status of Women in Computing Research (CRA-W), specifically through the Distributed Research Experiences of Undergraduates (DREU) Program.

## References

1. Mejino Jr, Jose LV, et al. "Human Development Domain of the Ontology of Craniofacial Development and Malformation."
2. L.G. Shapiro, E. Chung, T. Detwiler, J.L.V. Mejino(Jr), A.W. Agoncillo, J.F. Brinkley, and C. Rosse. Processes and problems in the formative evaluation of an interface to the foundational model of anatomy knowledge base. *J Am Med Inform Assoc.*, 12:35–46, 2005.
3. Hochheiser H, Aronow B, Artinger K, Beaty TH, Brinkley JF, Chai Y, Clouthier D, Cunningham ML, Dixon M, Donahue LR, Fraser S, Hallgrimsson B, Iwata J, Kelin O, Marazita ML, Murray JC, Murray S, de Villena FPM, Postlethwait J, Potter S, Shapiro L, Spritz, R, Visel A, Weinberg S and Trainor P. The FaceBase Consortium: A Comprehensive Program to Facilitate Craniofacial Research," *Developmental Biology*, Vol. 355, No. 2, 2011, pp. 175-182.
4. Clarkson, Melissa. "The Shirt Ontology." *The Shirt Ontology*. University of Washington, n.d. Web. 31 July 2013.
5. Brinkley, James F., MD, PhD, and Landon T. Detwiler, MS. "A Query Integrator and Manager for the Query Web ." *Journal of Biomedical Informatics* 45.5 (2012): 975-91. *SIG*. University of Washington, 31 Mar. 2012. Web.
6. Detwiler, Landon T., et al. "A relation-centric query engine for the foundational model of anatomy." *Medinfo* 11.Pt 1 (2004): 341-5.