OWL Paths

A library for processing SPARQL-like property paths over OWL classes

Melissa Clarkson
mclarkson@uky.edu

Todd Detwiler
todd.detwiler@uky.edu

University of Kentucky
Lexington, Kentucky, USA

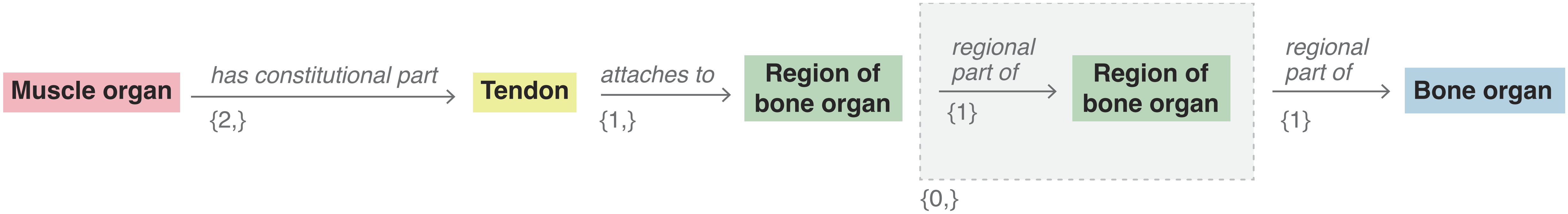
OWL Paths creation was motivated by our group’s need to validate consistency of modeling in a large ontology for human anatomy

Anatomical structures repeated throughout the body (such as joints, muscles, and bones) should be modeled in similar and predictable ways

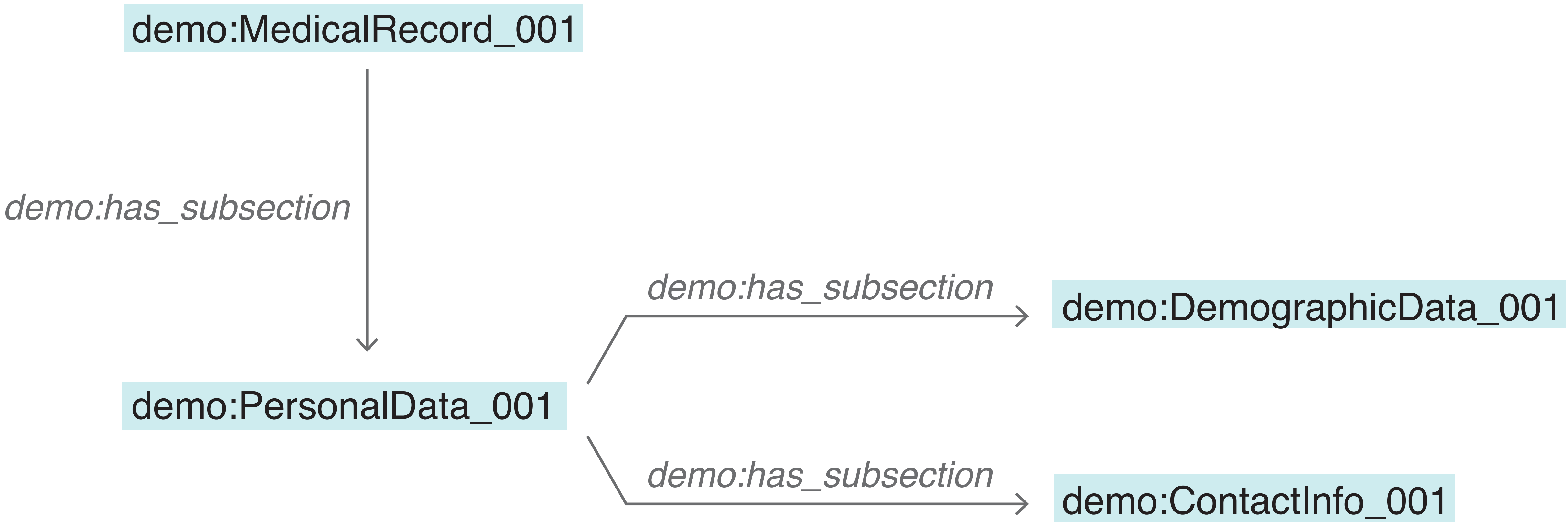
We want to use path expressions to verify modeling consistency in our ontology.

Our work is about writing rules for authoring and auditing large biomedical ontologies

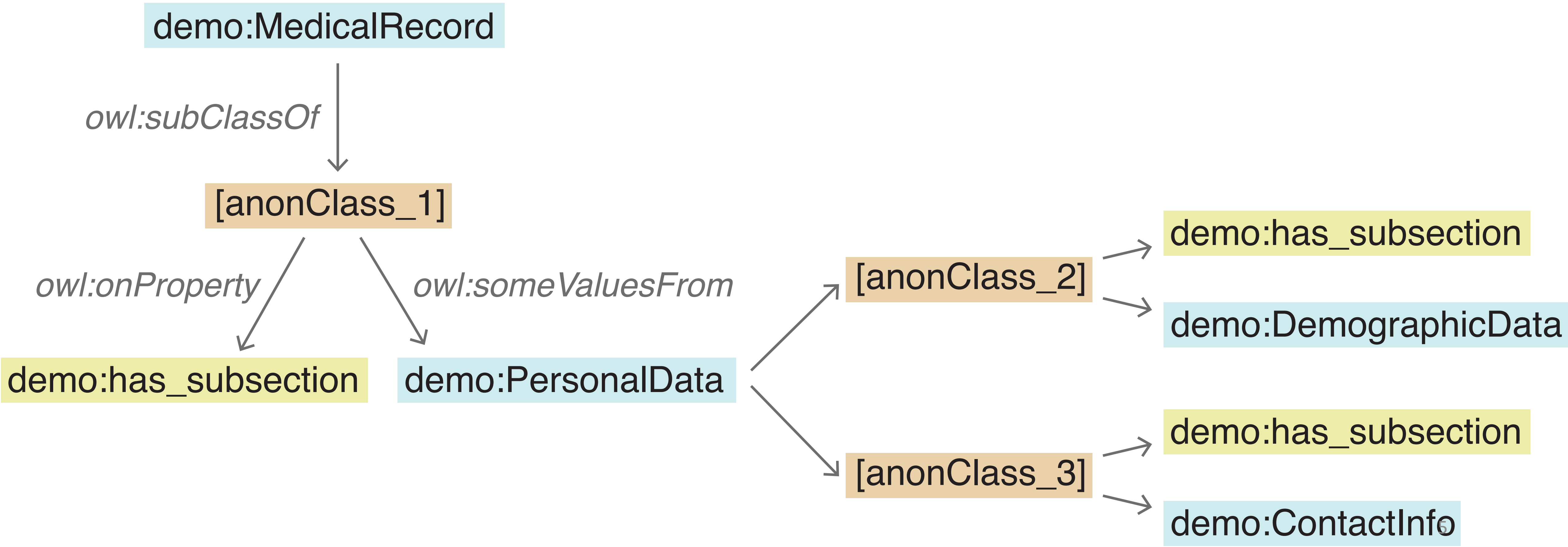
We call these rules "knowledge representation patterns"



A fictional medical record for an **individual** has these three triples:

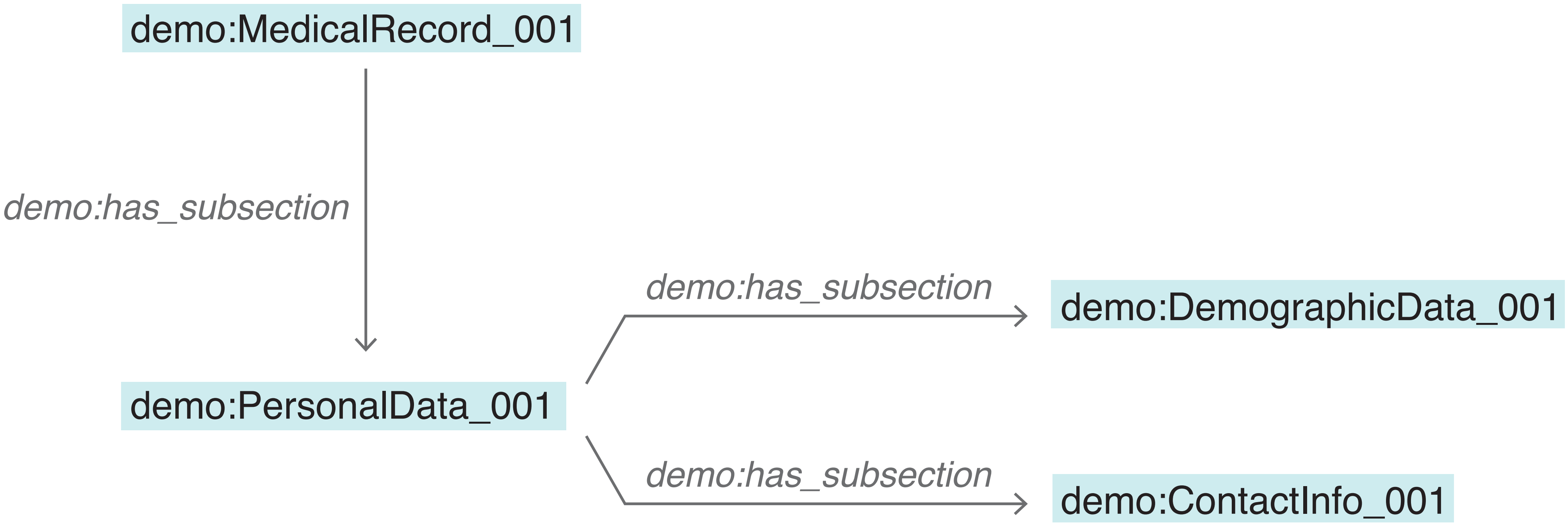


A fictional medical record **class** has this modeling:

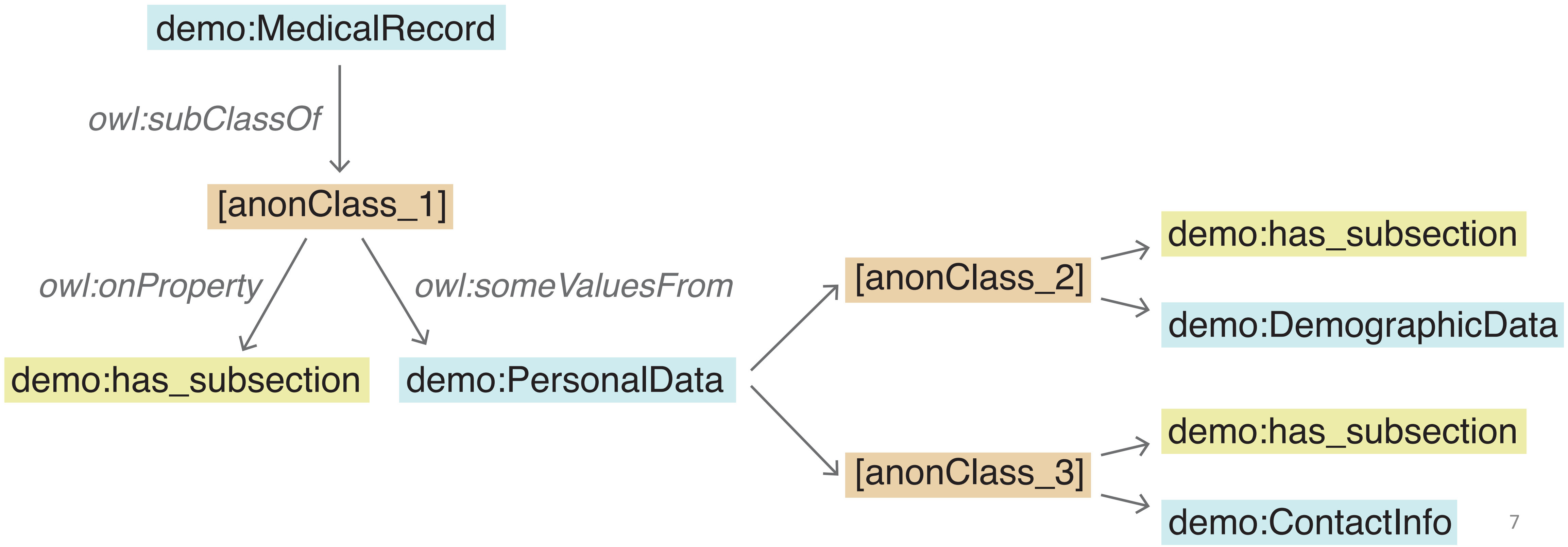


This fictional medical record for an **individual** can be queried by SPARQL for subsections

demo:MedicalRecord_001 demo:has_subsection+ ?x



But – SPARQL does not provide a way to query for classes whose individuals would be connected by a recursive property pattern



But – SPARQL does not provide a way to query for classes whose individuals would be connected by a recursive property pattern

- In practice this limitation is overcome by punning (a pre-processing step), and then using SPARQL
- The purpose of OWL Paths is to enable SPARQL-like path processing over the classes in an ontology as if the pun individuals were present

Representing anatomy

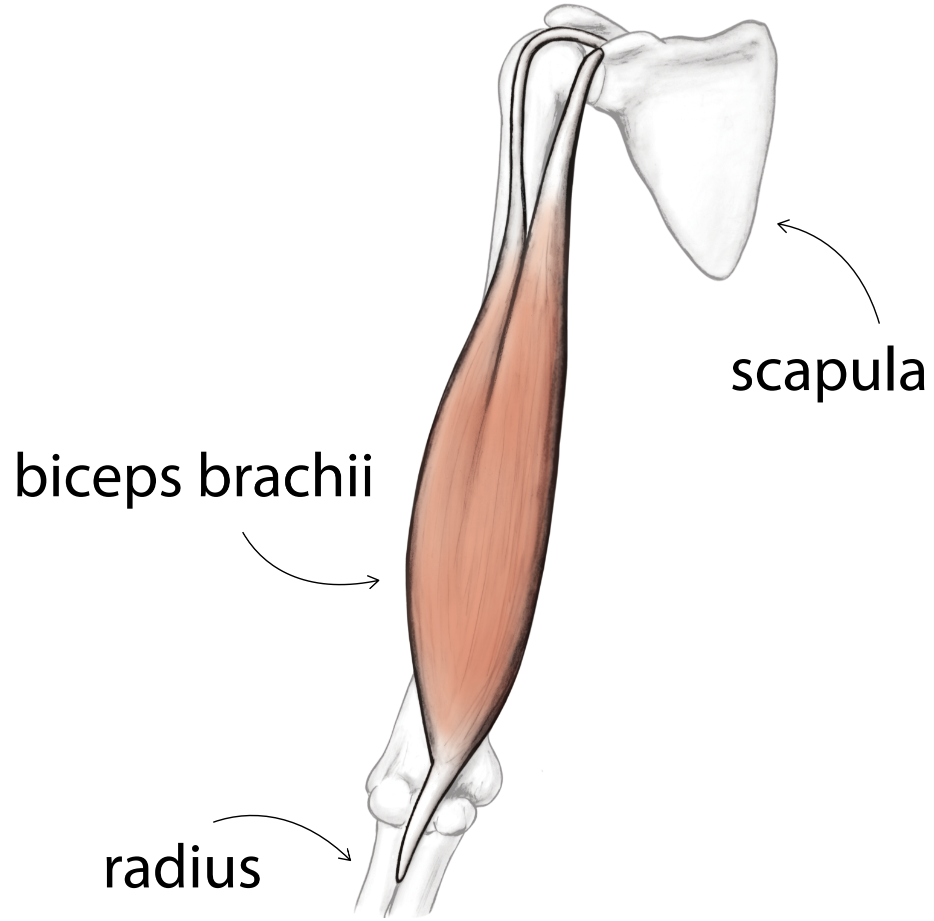
The anatomy ontology that catalyzed this work is the Foundational Model of Anatomy (FMA) ontology

- The FMA models adult anatomy using over 100,000 classes and 100 relations.
- In development for 20 years by the Structural Informatics Group at the University of Washington

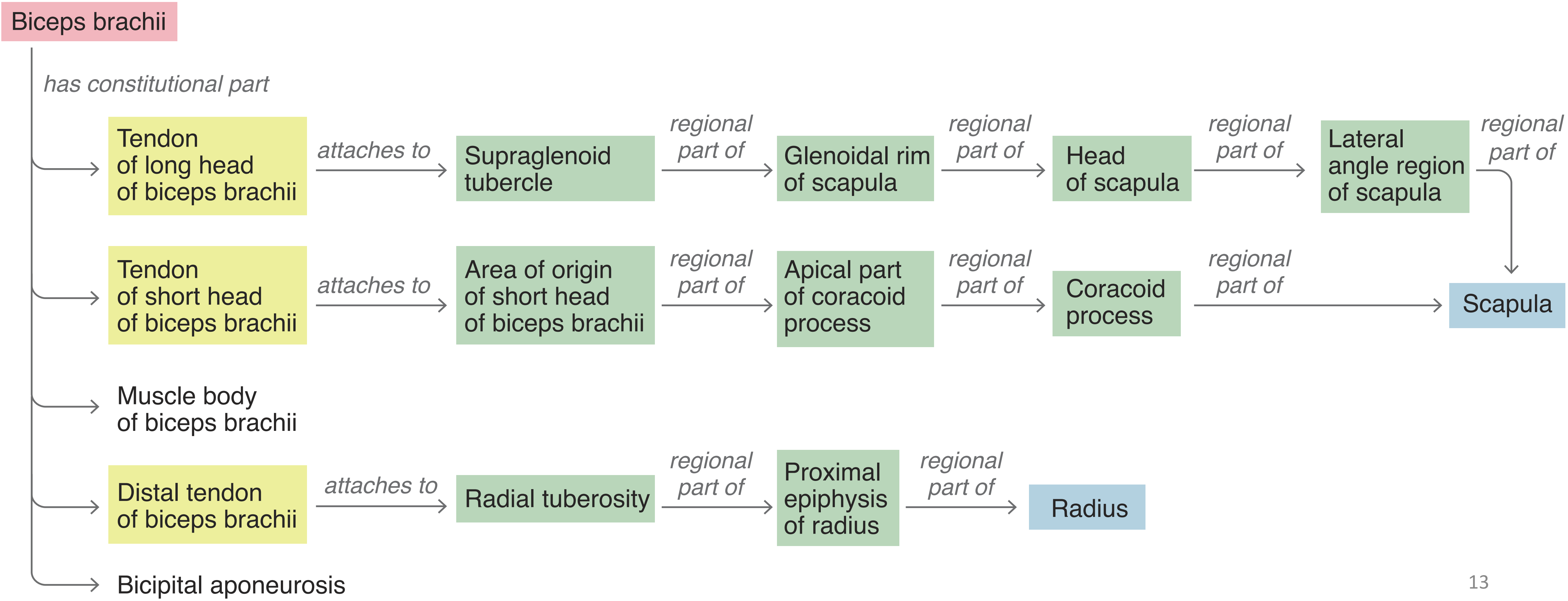
The anatomy ontology that catalyzed this work is the Foundational Model of Anatomy (FMA) ontology

- I have recently been funded to create a derivative called the Foundational Model of Human anatomy (FMHA)
- The FMHA is intended to be more consistent and complete in modeling than the FMA, and eventually to replace the FMA
- We will develop and implement knowledge representation patterns to construct and audit the FMHA

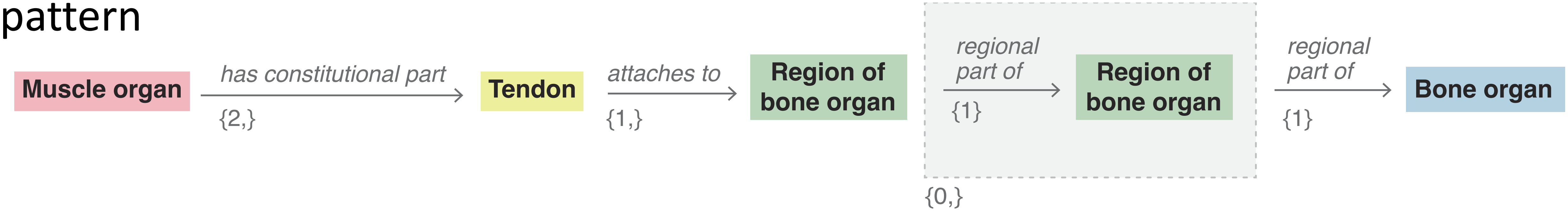
An example modeling a muscle (biceps brachii) and its relation to bones (scapula, radius)



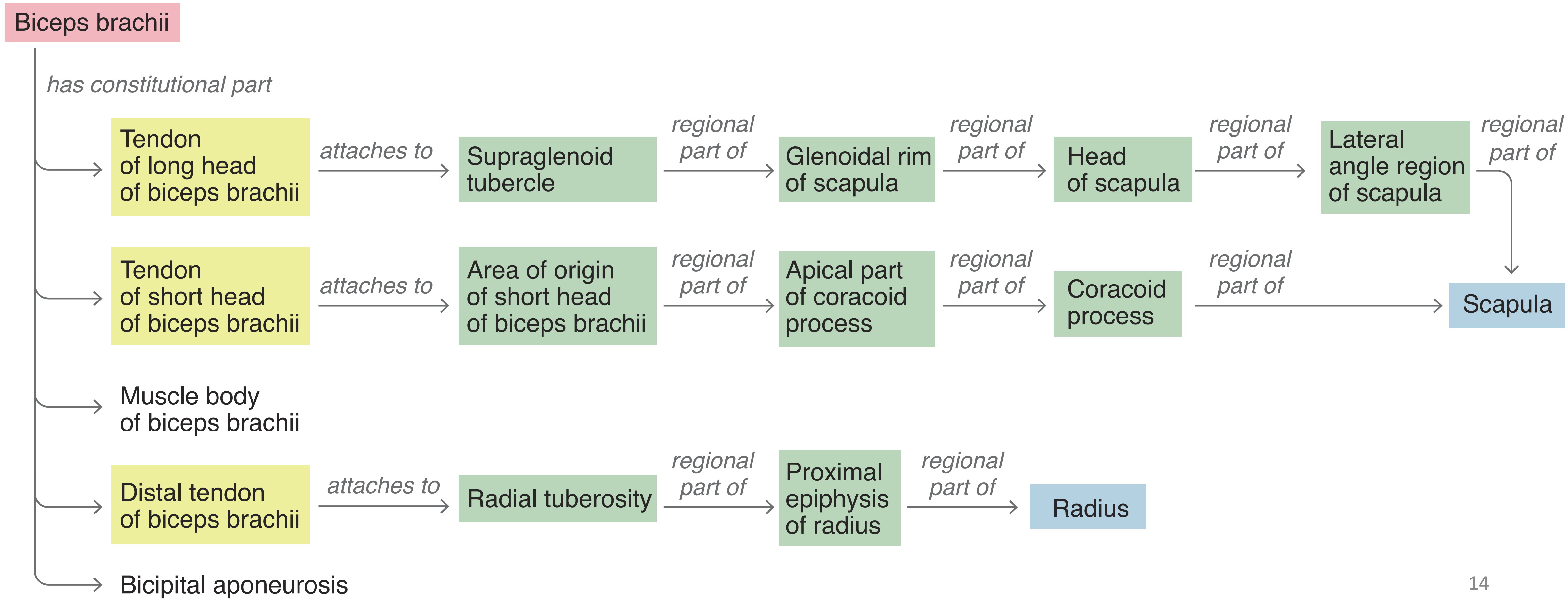
An example modeling a muscle (biceps brachii) and its relation to bones (scapula, radius)



pattern



ontology



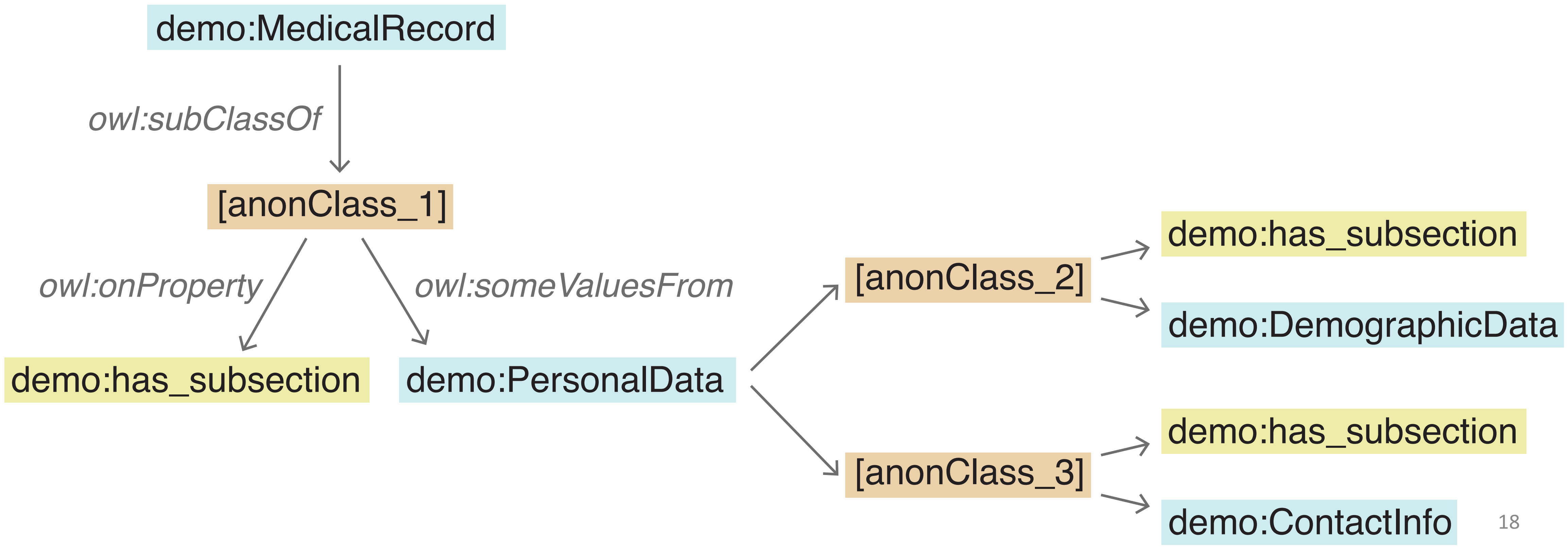
Implementation

The OWL Paths is a Java library built on the OWL API for processing property paths over OWL classes, in a similar manner as SPARQL is used to query using paths over OWL individuals

- OWL Paths grammar is syntactically similar to SPARQL
- Many but not all SPARQL constructs are supported
- The expressivity of OWL Paths needn't be limited to that of SPARQL

Syntax form	Matches	SPARQL	OWL Paths
uri	A URI or a prefixed name. A path of length one.	✓	✓
^elt	Inverse path (object to subject).	✓	
(elt)	A group path elt, brackets control precedence.	✓	✓
elt1 / elt2	A sequence path of elt1, followed by elt2	✓	✓
elt1 ^ elt2	Shorthand for elt1 / ^elt2, that is elt1 followed by the inverse of elt2.	✓	
elt1 | elt2	A alternative path of elt1, or elt2 (all possibilities are tried).	✓	✓
elt*	A path of zero or more occurrences of elt.	✓	✓
elt+	A path of one or more occurrences of elt.	✓	✓
elt?	A path of zero or one elt.	✓	
elt{n,m}	A path between n and m occurrences of elt.	✓	
elt{n}	Exactly n occurrences of elt. A fixed length path.	✓	
elt{n,}	n or more occurrences of elt.	✓	

We can use the library to query over classes whose individuals would include paths containing zero or more sequential occurrences of an object property



This is the code to set up the query and return the classes for the three subsections

```
// set up subjects
Set<OWLClassExpression> subjects = new HashSet<>();
OWLClass start = factory.getOWLClass( iri: "https://purl.org/owl/demo/medicalrecords.owl#MedicalRecord");
subjects.add(start);

// process OWL paths
String path = "demo:has_subsection*";
PathExpression pe = new PathExpression(structural_reasoner);
Set<OWLClassExpression> results = pe.processPath(path,subjects);
```

In summary – just as SPARQL enables property path exploration between **individuals**, this library is a utility that enables similar exploration between **classes**.

We are using this processing capability to validate **knowledge representation patterns** in a large biomedical ontology.

Source code at
https://gitlab.com/endless-forms-studio/owl_paths/

Melissa Clarkson
mclarkson@uky.edu

Todd Detwiler
todd.detwiler@uky.edu