Darwin-SW: Darwin Core-based terms for expressing biodiversity data as RDF

Editor(s): Guest Editors Semantics for Biodiversity
Solicited review(s): Mark Schildhauer and anonymous

Steven J. Baskauf [a,*] and Campbell O. Webb [b]
[a] *Department of Biological Sciences, PMB 351634, Vanderbilt University, Nashville, TN 37146, USA*
E-mail: steve.baskauf@vanderbilt.edu
[b] *Arnold Arboretum of Harvard University, 1300 Centre St., Boston 02131, USA*
E-mail: cw@camwebb.info

Abstract. Darwin-SW (DSW) is an RDF vocabulary designed to complement the Biodiversity Information Standards (TDWG) Darwin Core Standard. DSW is based on a model derived from a community consensus about the relationships among the main Darwin Core classes. DSW creates a new class to accommodate important aspects of its model that are not currently part of Darwin Core: a class of Tokens, which are forms of evidence. DSW uses Web Ontology Language (OWL) to make assertions about the classes in its model and to define object properties that are used to link instances of those classes. A goal in the creation of DSW was to facilitate consistent markup of biodiversity data so that RDF graphs created by different providers could be easily merged. Accordingly, DSW provides a mechanism for testing whether its terms are being used in a manner consistent with its model. Two transitive object properties enable the creation of simple SPARQL queries that can be used to discover new information about linked resources whose metadata are generated by different providers. The Organism class enables semantic linking of biodiversity resources to vocabularies outside of TDWG that deal with observations and ecological phenomena.

Keywords: Linked Data, Semantic Web, metadata, model, informatics

1. Introduction

Biodiversity data, records of the occurrences of organisms in their environment, have traditionally been recorded along with collected specimens preserved in natural history collections. Many specimen-based data have now been exposed electronically by a large number of providers. These data have been aggregated on a large scale and are augmented by observations data and data collected by newer mechanisms such as remote sensing and digital imaging. Expressing these data in the Resource Description Framework (RDF)[1] has a number of advantages over simple fielded text formats (where a line in the file contains the data for a single record). Because of its graph-based syntax,[2] RDF supports the complex data structures required to merge diverse kinds of data about biodiversity resources. RDF's use of triples as the basic unit of information removes ambiguity about the resource with which a property is associated, a fundamental problem when data about several types of resources are combined in a single row of a database table. RDF's use of URIs as globally unique identifiers (GUIDs) allows references to resources described by other data providers and facilitates participation in the global Linked Data[3] effort. HTTP URIs meet the requirements for persistent identifiers laid out in the Biodiversity Information Standards (TDWG)[4] GUID Applicability Statement standard[5]

* Corresponding author
[1] http://www.w3.org/RDF/
[2] http://www.w3.org/TR/rdf-syntax-grammar/
[3] http://linkeddata.org/
[4] http://www.tdwg.org/
[5] http://www.tdwg.org/standards/150/

which also specifies that GUIDs should resolve to return RDF. Readily available tools such as triple stores and the SPARQL query language[6] facilitate data aggregation and exploration of a composite database. Expressive extensions of RDF such as RDF Schema (RDFS)[7] and Web Ontology Language (OWL)[8] introduce the possibility of conducting machine reasoning on aggregated data and create the potential for discovery of new information that was not apparent in the separate data sources, a goal of the Semantic Web.[9] However, progress towards exposing biodiversity data as RDF has been hampered by the lack of a consensus graph-based model for the biodiversity domain.

There is much to draw on in the development of such a graph model. An early attempt to develop a standard model the biodiversity domain was the Association of Systematics Collections (ASC) Information Model for Biological Collections,[10] developed in 1992. Although the ASC model was a database model that predated RDF, it established many of the key biodiversity-related classes and mapped how they were related[11].

Darwin Core [1] is a general-purpose vocabulary designed to facilitate the transfer and integration of biodiversity data. It was developed from 1998 through 2009, when it was ratified as a TDWG technical specification standard.[12] Darwin Core consists of class and property terms, but does not assume a particular serialization nor does it endorse a particular data model. Existing guides describe how data can be exchanged as flat text files or as XML. Over time, Darwin Core has become the predominant standard for exchange of biodiversity data with over 428 million Darwin Core records aggregated by the Global Biodiversity Information Facility (GBIF) network as of 29 Jan 2014.[13]

In 2006, the Taxonomic Databases Working Group (TDWG; now Biodiversity Information Standards) began development of a high level technical architecture for information transfer. A central part of this architecture was the creation of an ontology to facilitate the integration of standards.[14] The result, written in OWL, was known as the TDWG Ontologies.[15] The TDWG Ontologies were never completed, were never widely implemented [2], and lack of funding made their maintenance difficult.[16]

The TaxonConcept ontology[17] was developed from 2009 to 2012 by Peter DeVries to link species names, concepts, and specimens using Linked Open Data (LOD) principles. It was not based specifically on TDWG terms but contained classes analogous to those in the TDWG ontologies.[18]

From 2012 to 2013, a team of writers from the TDWG RDF/OWL Task Group[19] drafted an RDF guide[20] for the Darwin Core standard. It provided guidelines for using Darwin Core property terms as RDF predicates. However, even with the RDF guide, Darwin Core does not provide object properties to link instances of its main classes (those classes that are not considered auxiliary terms).

This paper describes Darwin-SW ("SW" for Semantic Web; abbreviated DSW),[21] a vocabulary based on Darwin Core and designed to facilitate the description of biodiversity instance data as RDF. It was developed in 2010-11 to meet an immediate need in the Bioimages[22] and Xmalesia[23] projects to appropriately implement HTTP URI GUIDs, return meaningful RDF metadata when those URIs were dereferenced, and to provide a framework to which non-traditional resources, such as media items and ecological characteristics, could be linked. Following an extensive review of discussions on the TDWG email discussion list[24] an attempt was made to document the consensus about the meaning of existing Darwin Core classes and their relationships to each other.[25] Based on this perceived consensus, Darwin-SW was created using terms from the Darwin Core

[6] http://www.w3.org/TR/sparql11-overview/
[7] http://www.w3.org/TR/rdf-schema/
[8] http://www.w3.org/TR/owl-ref/
[9] http://www.w3.org/standards/semanticweb/
[10] http://wiki.tdwg.org/twiki/bin/viewfile/TAG/HistoricalDocuments?rev=1;filename=Ascmodrpt.pdf
[11] http://wiki.tdwg.org/twiki/bin/viewfile/TAG/HistoricalDocuments?rev=1;filename=Ascfig2.pdf
[12] http://www.tdwg.org/standards/450/
[13] http://www.gbif.org/
[14] http://wiki.tdwg.org/twiki/pub/TAG/TagMeeting1Report/TAG-1_Report_Final.pdf
[15] http://wiki.tdwg.org/twiki/bin/view/TAG/TDWGOntology
[16] http://www.hyam.net/blog/archives/643
[17] http://www.taxonconcept.org/
[18] http://code.google.com/p/tdwg-rdf/wiki/BiodiversityOntologies
[19] http://code.google.com/p/tdwg-rdf/
[20] The draft guide is at http://code.google.com/p/tdwg-rdf/wiki/DwcRdf with an eventual permanent URL of http://rs.tdwg.org/dwc/terms/guides/rdf/
[21] Current version at http://purl.org/dsw/; development and documentation at http://code.google.com/p/darwin-sw/ . Although based on Darwin Core, Darwin-SW has no official standing with TDWG.
[22] http://bioimages.vanderbilt.edu/
[23] http://xmalesia.info/
[24] http://code.google.com/p/darwin-sw/wiki/TdwgContentEmailSummary
[25] http://code.google.com/p/darwin-sw/w/list

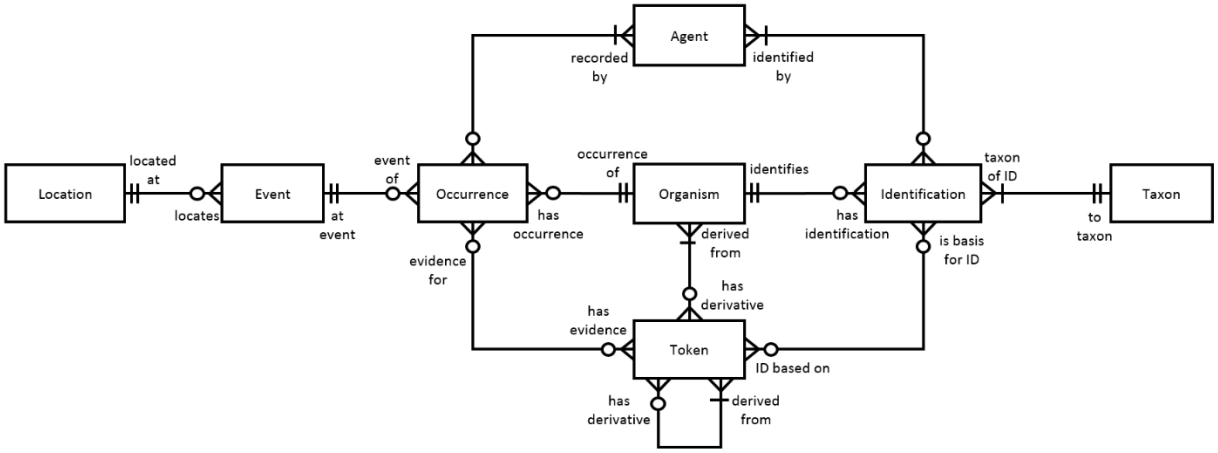

Fig. 1. Entity-relationship diagram of the Darwin-SW model using crow's foot notation with classes and relationships described in English text

standard and additional properties and classes defined within the DSW vocabulary.

In this paper, class names are capitalized when referenced in English text, e.g. Organism. In diagrams and tables, URIs are abbreviated using standard QNames for namespace prefixes, e.g. `dwc:Organism`. A list of prefixes and their corresponding URIs are listed in Table 3 of in the Appendix. URIs, RDF serializations, and SPARQL queries are written in `Courier` font. In many cases, URIs identify real resources in the wild, although example triples composed of those URIs are not necessarily asserted there. All RDF graph examples are serialized as Terse RDF Triple Language (Turtle).[26] All query examples use the SPARQL query language.

Section 2 of this paper describes the model, design considerations, and features of Darwin-SW. Sections 3 and 4 provide examples showing how DSW facilitates simple forms of reasoning and integration across biodiversity domains. Section 5 describes how DSW may evolve with Darwin Core in the future.

2. The Darwin-SW model

2.1. Design considerations

There has been longstanding interest in the TDWG community in leveraging Linked Data and Semantic Web technologies. The barrier caused by the lack of a consensus set of predicates for expressing biodiversity data as RDF would be lowered most easily if most of those predicates originated from terms already in common use. Because of the widespread acceptance of the Darwin Core standard, the choice was made to base Darwin-SW primarily on existing Darwin Core classes and the properties organized under them. The "lingua franca" status of Darwin Core makes integration of triples from multiple institutions more feasible.

Instances of key classes within the biodiversity domain (such as organisms, specimens, and taxa) within or among institutions can be linked using DSW-defined object properties that have no analogues in Darwin Core. Those instances can serve as anchor points to which resources outside the biodiversity community may be linked. Because of this cross-institutional and cross-community anchoring function, DSW expects that most resources will be identified by persistent HTTP URIs or HTTP-proxied versions of globally unique identifiers in accordance with the TDWG GUID Applicability Statement Standard.

Since the primary objective of DSW is to facilitate the linking of real data, and also since some Darwin Core classes have been applied with a wide variety of meaning, DSW takes a practical approach to the definition of classes. DSW sees class instances as nodes that group related properties rather than as entities that are heavily constrained ontologically (see sections 2.2.1 through 2.2.3 for specific examples). This approach differs significantly from that taken in the

[26] http://www.w3.org/TeamSubmission/turtle/

development of more formal ontologies such as the Biological Collections Ontology (BCO)[27] [3].

Similarly, although DSW uses terms from OWL in its definitions, it is not an ontology designed to enable extensive reasoning based on a hierarchical class structure. Nevertheless, the structure of DSW and properties assigned to its terms facilitate a number of simple but useful reasoning tasks which can be performed using SPARQL queries (see section 3 for examples).

2.2. Classes of the Darwin-SW model

The general form of the Darwin-SW model (Fig. 1) evolved from a discussion on the TDWG email discussion list[28] with an outline of the model suggested by Richard Pyle.[29] The model uses the main classes of Darwin Core. Sections 2.2.1. through 2.2.3. provide additional details about key classes in the model. The class relationships were influenced[30] by the ASC model which included analogues of the Location, Event, Identification, Taxon, Agent, and Reference classes.

2.2.1. Organism

A key addition of the DSW model was the inclusion of an Organism class. The existence of this class was implied by the existing Darwin Core term `dwc:individualID` which was defined as "An identifier for an individual or named group of organisms represented in the Occurrence. Meant to accommodate resampling of the same individual or group for monitoring purposes." The definition of `dwc:individualID` suggested features that were incorporated in the DSW concept of Organism. An Organism is not restricted to a single biological organism: it can be any sort of organism, clone, colony, or group of organisms that is typically observed or sampled over time. An additional requirement established during the email discussion was that Organism instances should be taxonomically homogeneous because they would be the objects of taxonomic determinations (i.e. Identification instances). A third feature of an Organism is that serves as an anchor point for resources derived from it such as specimens, images, and samples. So although an Organism can be described in conceptual terms by comparison with biological organisms, from the standpoint of the DSW model, an Organism is a node that connects Occurrences, Identifications, and derived resources [4]. On 2014-10-26, the `dwc:Organism` class, which has the features described in this section, was added to Darwin Core and the term `dwc:individualID` was deprecated in favor of `dwc:organismID`.[31] To maintain consistency with Darwin Core, the Darwin-SW-minted term for this class was deprecated in favor of `dwc:Organism`.

2.2.2. Occurrence

The Darwin Core Occurrence class (`dwc:Occurrence`) originally functioned effectively as a superclass of specimens and observations. However, in the email discussion it was clear that many participants favored a more abstract notion of Occurrence as an organism at a time and place. This notion resulted in the placement of Occurrence between Organism and Event in the DSW model. On 2014-10-26, the definition of `dwc:Occurrence` in the Darwin Core standard was changed to reflect the more abstract notion. A more exact definition of Occurrence that places it in an OBO-style[32] hierarchy is not necessary in order for an Occurrence instance to perform its linking function.

2.2.3. Taxon Concepts

Taxa are an important component of any biodiversity model and significant effort has been expended towards defining the meaning of the terms "taxon" and "taxon concept" [5]. The TDWG Taxon Concept Transfer Schema (TSC) standard[33] describes taxon concepts as a name plus an "according to" statement that provides information about the reference that circumscribes the taxon. The definition of the Darwin Core Taxon class and its associated comment adopted on 2014-10-26 implies that that a `dwc:Taxon` instance is a taxon concept in the sense of the TCS standard. DSW accepts this view of taxa and links Identification instances to Taxon instances using the term `dsw:toTaxon`. The draft Darwin Core RDF guide specifies a new term, `dwciri:toTaxon` that serves the same purpose, and DSW will use that term instead of `dsw:toTaxon` when the Guide is adopted.

[27] http://purl.obolibrary.org/obo/bco.owl
[28] http://lists.tdwg.org/pipermail/tdwg-content/
[29] http://lists.tdwg.org/pipermail/tdwg-content/2010-October/001703.html
[30] http://lists.tdwg.org/pipermail/tdwg-content/2010-October/001718.html
[31] http://rs.tdwg.org/dwc/terms/history/decisions/
[32] http://www.obofoundry.org/
[33] http://www.tdwg.org/standards/117/

Table 1

Object properties that link the main classes in the Darwin-SW model.

Object property	Subject class (domain)	Object class (range)
dsw:locates	dcterms:Location	dwc:Event
dsw:locatedAt	dwc:Event	dcterms:Location
dsw:eventOf	dwc:Event	dwc:Occurrence
dsw:atEvent	dwc:Occurrence	dwc:Event
dsw:occurrenceOf	dwc:Occurrence	dwc:Organism
dsw:hasOccurrence	dwc:Organism	dwc:Occurrence
dsw:hasIdentification	dwc:Organism	dwc:Identification
dsw:identifies	dwc:Identification	dwc:Organism
dsw:toTaxon	dwc:Identification	dwc:Taxon
dsw:taxonOfId	dwc:Taxon	dwc:Identification

Each property has an inverse property linked by an owl:inverseOf relationship. For each property, the intended subject class is declared the rdfs:domain of the term and the intended object class is declared the rdfs:range of the term. Namespace prefixes are defined in Table 3 of the Appendix. Upon adoption of the Darwin Core RDF Guide, dsw:toTaxon will be deprecated in favor of dwciri:toTaxon.

Table 2

Evidence-related object properties in the Darwin-SW model.

Object property	Subject class	Object class
dsw:derivedFrom	any kind of derived resource	dwc:Organism or any kind of derived resource
dsw:hasDerivative	dwc:Organism or any kind of derived resource	any kind of derived resource
dsw:hasEvidence	dwc:Occurrence	dwc:Token
dsw:isEvidenceFor	dwc:Token	dwc:Occurrence
dsw:idBasedOn	dwc:Identification	dsw:Token
dsw:isBasisForId	dsw:Token	dwc:Identification

Each property has an inverse property linked by an owl:inverseOf relationship. dsw:derivedFrom and dsw:hasDerivative do not have declared ranges or domains. For the other four properties, the intended subject class is declared the rdfs:domain of the term and the intended object class is declared the rdfs:range of the term. Namespace prefixes are defined in Table 3 of the Appendix.

2.3. Object properties in Darwin-SW

Darwin-SW defines a number of object properties used to link classes in the DSW model (Tables 1 and 2). In most cases, these properties occur in pairs that are related by an owl:inverseOf property. By providing pairs of object properties, DSW allows a resource in either linked class to serve as the subject of the triple that provides the linkage. A reasoner can infer the alternate linking triple whose predicate is the corresponding inverse property if the provider does not assert that triple directly. This inverse pair strategy permits less verbose use by data creators outside the traditional Occurrence-centric or Taxon-centric biodiversity informatics domain.

Table 1 shows the object properties defined by DSW to link its main classes (Location, Event, Occurrence, Organism, Identification, and Taxon Concept). Because the primary objective of DSW is to facilitate the linking of real data, these object properties serve primarily as a means to facilitate one-to-many or many-to-many relationships among instances of the main classes.

2.3.1. Properties linking to Agents

In addition to the object properties intended to link its main classes, DSW also defines the object properties dsw:georefBy, dsw:recBy, and dsw:idBy which have non-literal ranges (foaf:Agent) and are analogues of the Darwin Core literal value terms dwc:georeferencedBy, dwc:recordedBy, and dwc:identifiedBy.

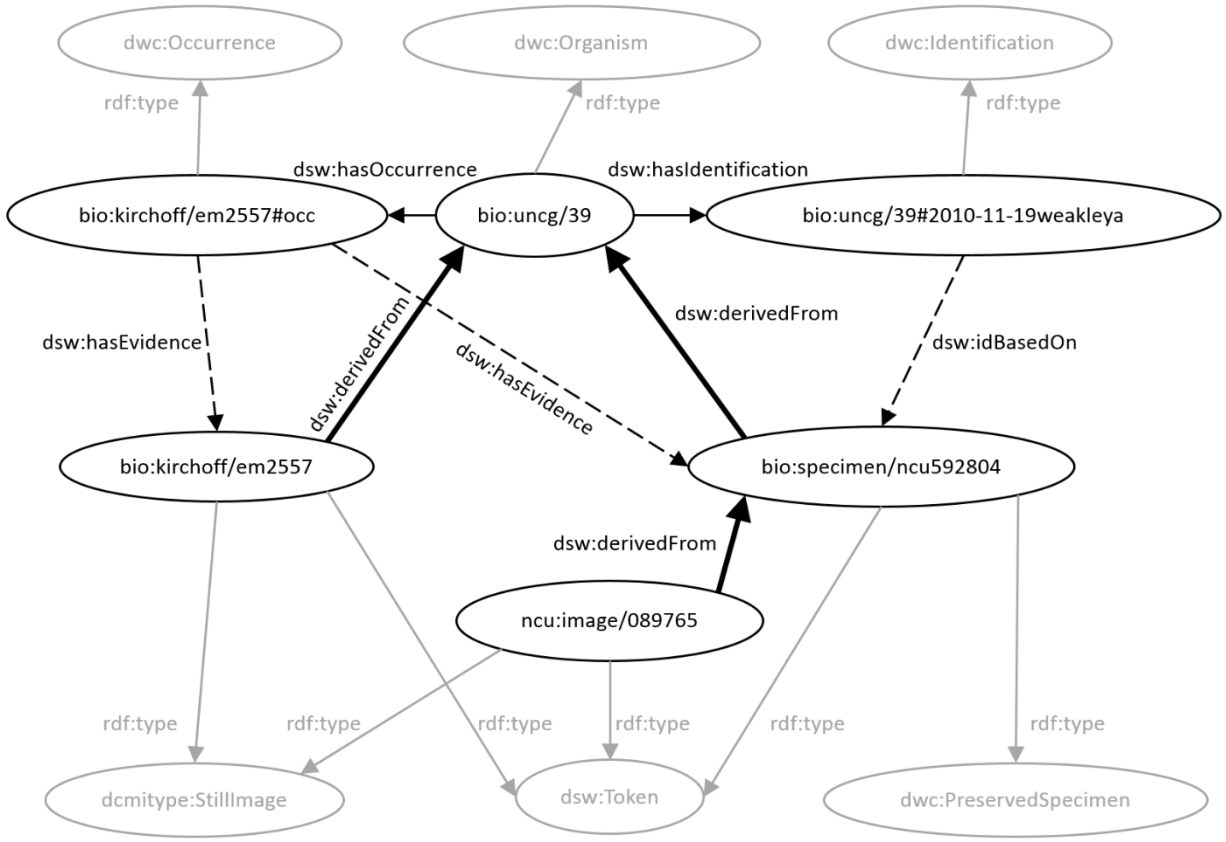

Fig. 2. Properties used to link evidence to key classes in the Darwin-SW model. Bold arrows are transitive `dsw:derivedFrom` properties. Dashed arrows are object properties that link to supporting evidence. For clarity, inverse object properties are not shown. `rdf:type` assertions are shown in gray. `bio:` is an abbreviation for the real namespace `http://bioimages.vanderbilt.edu/`. `ncu:` represents a fake namespace that contains identifiers for specimen images. See Fig. 4 in the Appendix for a serialization of this graph.

When the Darwin Core RDF guide is ratified, these properties will be deprecated in favor of terms in the `dwciri:` (`http://rs.tdwg.org/dwc/iri/`) namespace as suggested in the guide.

2.3.2. Properties linking to evidence

A key innovation of Darwin-SW is the recognition of the role of evidence in documenting the occurrences and identifications of organisms. DSW recognizes organism-related evidence by explicitly defining the class `dsw:Token`. A Token is essentially a physical, digital, or conceptual voucher that provides some kind of evidence about an Organism. A Token may be part of the Organism or the Organism itself in living or preserved form (i.e. as a specimen). It may also be an image, sound, sample, DNA sequence, or human or machine observation.

Tokens are linked to the Organism from which they originated by the transitive object property `dsw:derivedFrom` and its inverse `dsw:hasDerivative` (Table 2, Fig 2). Thus `dsw:derivedFrom` serves as a mechanism for documenting the provenance of diverse resources used as evidence. A wide variety of types of resources can serve as evidence and most Token instances of will have one or more additional `rdf:type` declarations. There is also an expectation that the Token or at least the metadata about that Token will be preserved and made accessible to anyone who may want to verify the assertions made

about the Organism, its Occurrences, and its Identifications.

DSW defines two pairs of object properties that can be used to indicate the evidence on which an assertion about an occurrence is based (Table 2; Fig. 2). `dsw:hasEvidence` and `dsw:evidenceFor` link an Occurrence to a Token that vouches for the recording of the Occurrence. Several forms of evidence (e.g. images, material samples, and specimens) may be archived to document a single Occurrence. In this way, the time and place metadata (inherent in the Occurrence) for the Tokens can be efficiently shared. `dsw:idBasedOn` and `dsw:isBasisForId` link an Identification to a Token that was used to make a taxonomic determination. Zero to many Tokens may be linked as the basis for an identification (with zero representing the case where there is no known token preserved to document the Identification).

3. Querying and reasoning facilitated by DSW

3.1. Denormalization and inconsistent term use

A design goal of Darwin-SW was to facilitate biodiversity data integration by making it feasible to merge graphs containing RDF data exposed by different providers. The combined graph could then be queried to discover information that would not be apparent through examination of the separate graphs.

One possible impediment to this kind of cross-institutional data integration is inconsistent use of object properties. In this section, the term "normalize" is used in the relational database sense (as opposed to the canonical normalized graph sense). Depending on the type of resource of interest to particular data providers, they may structure their non-RDF databases using different levels of normalization. When the data in those databases are exposed as RDF, providers may not include instances of classes that are included in the Darwin-SW model.

For example, providers that are interested in recording many Occurrences at an Event and many Events at a Location will create Event instances (Fig. 1). Using Darwin-SW, they can then link Occurrences to Events using `dsw:atEvent`, and link the Events to Locations using `dsw:locatedAt`. (Appendix, Example 1)

Other providers that are not interested in linking many Occurrences to a single Event may have denormalized their model to eliminate Event. Such providers may inappropriately link Occurrences directly to Locations using `dsw:locatedAt`. (Appendix, Example 2)

Although in the spirit of the semantic web providers may link the resources they describe in any manner, using Darwin-SW object properties to link class instances in a manner that is inconsistent with Fig. 1 and Table 1 is counterproductive to the design goal of enabling data integration and effective querying. A simple SPARQL query based on a graph pattern that assumed Occurrences were linked to Events and that Events were linked to Locations would fail to bind Occurrences that were linked directly to Locations. (Appendix, Query 1) Thus inconsistent use of DSW object properties caused by denormalizing the DSW model inhibits effective querying.

It would be possible to construct a more complex SPARQL query to accommodate alternate degrees of normalization. However, because there are six main classes in DSW, there would be many possible ways that denormalization could occur and it would be difficult to account for all possible permutations using any reasonable query. For this reason, DSW assumes the most normalized model that is likely to be of interest to the biodiversity informatics community. If necessary, providers can create blank nodes representing class instances that are not explicitly present in their database in order to use DSW object properties consistently with the DSW model. (Appendix, Example 3)

3.2. Detecting inconsistencies using ranges, domains, and disjoint classes in Darwin-SW

Because RDF allows anyone to say anything about anything, there is no simple way to enforce appropriate linking using DSW object properties. However, the properties of DSW terms make it possible to detect inconsistent links of the sort discussed in section 3.1.

Darwin-SW declares domains and ranges for most object properties that it defines (Tables 1 and 2). A client can infer types that are entailed by these domain and range declarations. Because DSW declares all of its main classes to be mutually disjoint, a client can detect inconsistent DSW object property use when an inferred type of a resource conflicts with a disjoint type that is explicitly declared for that same resource. An inconsistency can also be generated simply by using two DSW object properties in a manner inconsistent with the DSW model. For example, asserting:

```
ex:organism dsw:hasOccurrence ex:occurrence.
ex:occurrence dsw:locatedAt ex:location.
```

generates an inconsistency without any explicit type declarations since the range of `dsw:hasOccurrence` entails that `ex:occurrence` is an Occurrence while the domain of `dsw:locatedAt` entails that it is an Event.

Using this approach, graphs containing data from new providers could be screened for inappropriate use of object properties before they are merged permanently with an existing multi-institution graph. An OWL reasoner could detect the change of the multi-institution graph from consistent to inconsistent upon the addition of the incoming graph if the incoming graph contained DSW object properties used in a manner that conflicted with the DSW model. However, the limited reasoning that is required to screen for inappropriate linking can be done using several simple SPARQL queries (see Queries 2 and 3 in the Appendix). These queries are not computationally intensive and only one needs to be run on the entire merged dataset.

3.3. Cross-institutional discovery

The ability to discover previously unknown information that is entailed by asserted triples is one of the most compelling reasons for expressing data using RDF vocabularies instead of more traditional database methods.

The semantics imposed by Darwin-SW term definitions may entail some simple information not explicitly expressed. For example, an asserted triple:

```
<http://bioimages.vanderbilt.edu/uncg/39>
dsw:hasOccurrence
<http://bioimages.vanderbilt.edu/kirchoff/em2
557#occ>.
```

entails

```
<http://bioimages.vanderbilt.edu/kirchoff/em2
557#occ> dsw:occurrenceOf
<http://bioimages.vanderbilt.edu/uncg/39>.
```

because `dsw:hasOccurrence` and `dsw:occurrenceOf` are declared to be inverse properties.

Although it may be useful to generate unstated triples of this sort, that hardly constitutes "discovery" of novel information. However, DSW enables the discovery of truly novel information by making it possible to link in a simple and consistent manner data about resources originating from different institutions, conducting limited inferencing, and then querying the graph that is a merge of the inferred and asserted triples. In particular, use of the transitive object property `dsw:derivedFrom` (and its inverse, `dsw:hasDerivative`) make it possible to construct very simple queries that can discover information that would not be obvious to a single provider. This is illustrated for several important use cases in sections 3.3.1 through 3.3.3.

3.3.1. Linking duplicates

It is common practice in the collections community to collect multiple specimens from the same organism, colony, or local population (a taxonomically homogeneous entity and therefore a `dwc:Organism`). Such specimens are called "duplicates". They are often distributed to other institutions to safeguard records of occurrence and as a courtesy to help build collections. Linking of duplicates is an important use case that has been addressed by projects such as FilteredPush[34] and BiSciCol.[35]

Unfortunately, records of duplicate exchanges are often poor, particularly when the specimens are old. This creates a problem when new information is linked to one duplicate without the knowledge of other institutions holding duplicates.

Despite the colloquial meaning of the word "duplicate", it would be incorrect to assert equivalence between duplicate specimens since they are distinct entities with different post-collection histories. But because of the nature of the origin of duplicate specimens, it would be correct to state that each duplicate was derived from a single Organism, and it would be appropriate to link each duplicate to the same Organism instance using `dsw:derivedFrom`. Similarly, it would also be appropriate to link each duplicate to the same Occurrence instance using `dsw:evidenceFor` if the collection time and location of the duplicates were the same.

For example, if Provider 1 described a specimen like this:

[34] http://wiki.filteredpush.org/wiki/ActuallyFindingDuplicates2
[35] http://biscicol.blogspot.com/p/home.html

```
provider1:organism1 a dwc:Organism;
     dsw:hasOccurrence provider1:occ1.
provider1:occ1 a dwc:Occurrence.
provider1:spec1 a dwc:PreservedSpecimen;
     dsw:evidenceFor provider1:occ1;
     dsw:derivedFrom provider1:organism1.
```

Provider 2 could assert that another specimen was a duplicate using:

```
provider2:spec2 a dwc:PreservedSpecimen;
     dsw:evidenceFor provider1:occ1;
     dsw:derivedFrom provider1:organism1.
```

If the specimen were collected from the same Organism but as part of a different Occurrence (i.e. at a different time and possibly a different place), Provider 2 could create a different Occurrence instance while still asserting a `dsw:derivedFrom` relationship to the same Organism.

In a case where the duplicate status were initially unknown, it would be likely that each institution that databased one of the duplicate specimens would mint a separate URI for the Organism from which its specimen was derived. A later discovery that the specimens were duplicates would imply that the two Organism URIs were simply different identifiers for the same resource.

Two URIs identifying an identical resource can be linked using the predicate `owl:sameAs`. The semantics of `owl:sameAs`[36] are such that any statement made about a resource denoted by one URI is also true when made substituting a URI that has been linked to the first URI by `owl:sameAs`. Thus a reasoner can construct all missing triples that could be created by substituting one of the URIs for the other. This allows resources linked to either Organism URI to be bound in a query by triple patterns that would otherwise match only triples containing one URI or the other.

For example, Provider 3 might have described a specimen in its collection like this:

```
provider3:organism3 a dwc:Organism;
     dsw:hasOccurrence provider3:occ3.
provider3:occ3 a dwc:Occurrence.
provider3:spec3 a dwc:PreservedSpecimen;
     dsw:evidenceFor provider3:occ3;
     dsw:derivedFrom provider3:organism3.
```

If Provider 3 discovered later that the specimen was a duplicate documenting the same Occurrence as Provider 1's specimen, it could document that discovery by making the following assertions:

```
provider3:organism3
    owl:sameAs provider1:organism1.
provider3:occ3 owl:sameAs provider1:occ1.
```

If the specimen were collected from the same Organism, but at a different time or location, Provider 3 could indicate that by asserting only:

```
provider3:organism3
    owl:sameAs provider1:organism1.
```

Documenting duplicates in this way creates links that facilitate discovery of any new information related to the linked resources. For example, if a new taxonomic determination were made based on Provider 1's specimen:

```
provider1:organism1
    dsw:hasIdentification provider1:id2.
provider1:id2 dsw:idBasedOn provider1:spec1.
```

Providers 2 and 3 could discover this new information related to their specimens because their specimens were linked to the same Organism instance. The following section describes methods that can be used to make such discoveries.

3.3.2. Discovering new derived resources and modified metadata

Organisms and specimens have increasingly become the source of a variety of derived resources obtained through physical and electronic means. Tissue samples, DNA sequences, digital images, telemetry, digital sound recordings, and video may be generated directly from organisms or from resources derived from organisms. As they are generated, these resources may pass to new institutions, and new metadata about the resources may be created without the knowledge of holders of related resources.

The object property `dsw:derivedFrom` is used to link an organism-derived resource to its parent resource, which may be either an Organism or another organism-derived resource. In essence, `dsw:derivedFrom` provides generic way to track the provenance of organism-related resources by establishing links from them to each of their ancestral resources. Other class-specific terms such as `foaf:depicts` or `dcterms:isPartOf` can be used to establish more specific kinds of relationships with parent resources.

[36] http://www.w3.org/TR/owl-ref/#sameAs-def

In Fig. 2, the specimen image is linked to its parent specimen by the triple:

```
<http://herbarium.unc.edu/image/089765>
    dsw:derivedFrom
<http://bioimages.vanderbilt.edu/specimen/ncu
592804>.
```

and that specimen is linked to its parent Organism (a tree) by the triple:

```
<http://bioimages.vanderbilt.edu/specimen/ncu
592804> dsw:derivedFrom
<http://bioimages.vanderbilt.edu/uncg/39>
```

A reasoner can infer the triple

```
<http://herbarium.unc.edu/image/089765>
      dsw:derivedFrom
<http://bioimages.vanderbilt.edu/uncg/39>
```

which links the specimen image directly to its more distant ancestor (the tree) based on the transitivity of `dsw:derivedFrom`. Since a properly designed SPARQL query can construct triples for entailed but unexpressed `dsw:derivedFrom` relationships, those triples can be added to a large multi-institution graph efficiently by a client designed to conduct a limited set of reasoning tasks. In SPARQL 1.1, arbitrary length path matching can be used traverse chained `dsw:derivedFrom` links without materializing every possible entailed relationship, making it easier to take advantage of the transitivity.

One implication of the transitivity of `dsw:derivedFrom` is that a reasoner will infer a `dsw:derivedFrom` relationship from a resource to *every* ancestral resource from it up to the original Organism. Similarly, a reasoner will infer `dsw:hasDerivative` relationships from an Organism to *every* descendant resource that is linked to the Organism though a chain of explicit parent/child `dsw:hasDerivative` links. This creates a powerful tool for querying because once a reasoner has constructed triples for all entailed `dsw:derivedFrom` and `dsw:hasDerivative` relationships, it becomes a simple matter to conduct queries that apply to all derivatives of a particular Organism. For example, this simple query:

```
SELECT DISTINCT ?resource ?type
WHERE  {
 ?resource dsw:derivedFrom
<http://bioimages.vanderbilt.edu/uncg/39>.
 ?resource a ?type.
 ?resource dcterms:created ?date.
 FILTER(?date >= xsd:date ("2012-01-01"))
 }
```

would discover new resources derived from the Organism in Fig. 2 that were created after the beginning of 2012.

This is an uncomplicated illustration because it involves a single Organism with few `dsw:derivedFrom` links. However, variations on this approach provide a powerful way to discover less apparent information in situations that are more complex. For example, a provider of images of organisms or organism specimens would be interested in knowing if organisms from which its images were derived had been assigned new determinations. It would probably be unaware of those determinations if they were based on some evidence other than an image in its collection (e.g., based on specimens held elsewhere). This query:

```
  SELECT DISTINCT ?resource ?sciName
  WHERE  {
   ?resource dwciri:inCollection
  <http://biocol.org/urn:lsid:biocol.org:col
:15495>.
   ?resource a dcmitype:StillImage.
   ?resource dsw:derivedFrom ?individual.
   ?individual dsw:hasIdentification ?id.
   ?id dwc:scientificName ?sciname.
   ?id dwc:dateIdentified ?date.
 FILTER(?date >= xsd:date("2012-01-01"))
   }
```

would discover any new determinations that were made after the start of 2012 that were relevant to any images in the collection identified by the URI `http://biocol.org/urn:lsid:biocol.org:col:15495`. This would include determinations for duplicate specimens if they were linked as described in section 3.3.1. The image provider could discover a new determination made based on a duplicate specimen without the holder of the duplicate even knowing that the provider's image existed.

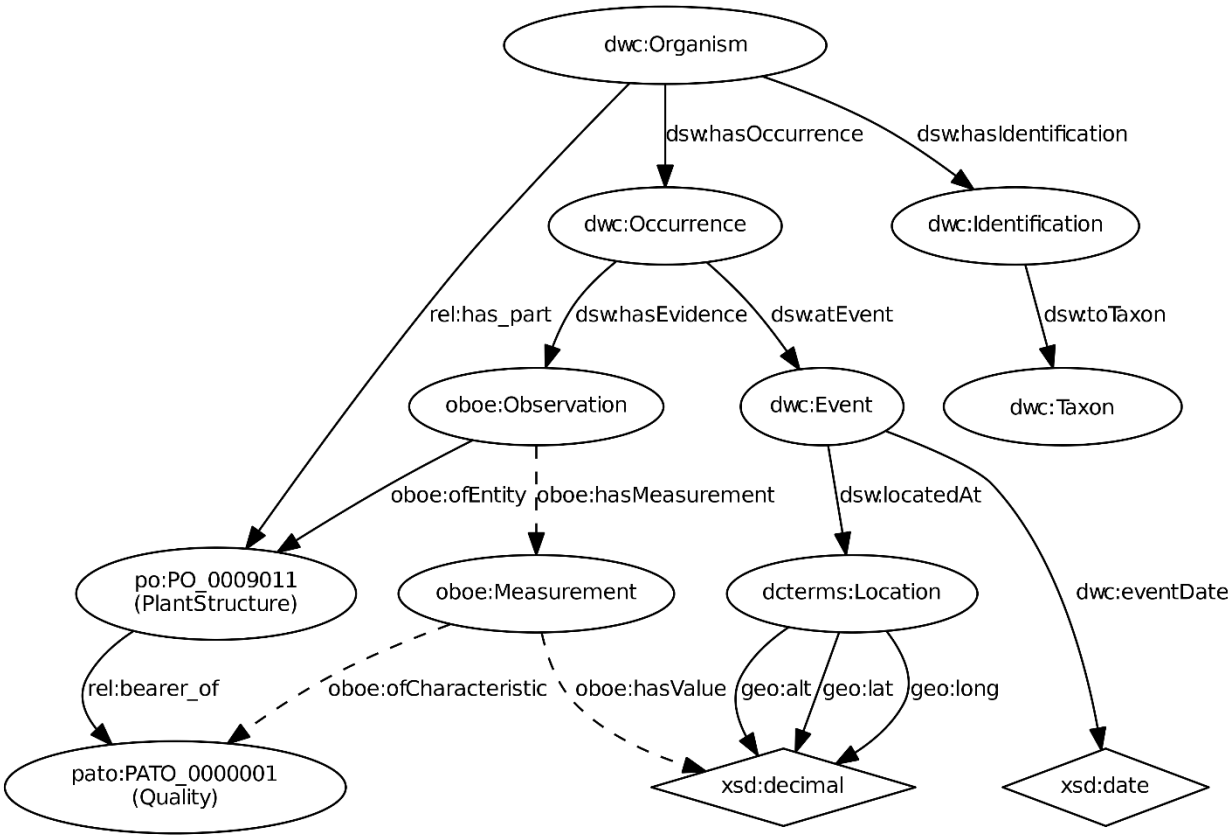

Fig. 3. Diagram of a graph structure that can be used to represent observations of organisms that also have identifications. Note that the diagram shows links between *instances* of classes, but for simplicity only the class URIs of those instances are indicated in the ovals. `rel:has_part` is an abbreviation for `http://purl.obolibrary.org/obo/BFO_0000051` and `rel:bearer_of` is an abbreviation for `http://purl.obolibrary.org/obo/RO_0000053`. See section 4 of the text for more details.

In these two examples, the chain of derived resources was relatively short. However, one can imagine realistic scenarios where the chain of derived resources was much longer. For example, consider a bird that is captured and banded by institution A. A tissue sample collected during the capture is sent to institution B which extracts and sequences DNA, with the sequence deposited in the repository of institution C. At a later time the bird is found dead and given to institution D which accessions the bird skin and associates the specimen with the earlier collection event based on the information on the band. Institution D loans the skin to Institution E, where an expert makes a determination based on comparison with other skins in the collection at Institution E. If each institution linked its resource to the parent resource from which the institution's resource was derived using `dsw:derivedFrom`, and if the graphs of metadata from all institutions were merged, a reasoner could link all descendant resources directly to the bird Organism instance. Institution C's DNA sequence repository could use a simple query similar to the examples in this section to discover the determination made at Institution E without tracking or even knowing about the intervening chain of resources between its sequence and the determination.

These examples illustrate how a relatively uncomplicated set of RDF properties combined with simple queries can discover kinds of information that would be very difficult to track using conventional database methods. The challenge therefore becomes more social than technological since the barriers to achieving that kind of discovery require community efforts such as adoption of standard vocabularies, commitment to persistent URI identifiers, reuse of identifiers assigned by others, and establishment of a consensus

triple store into which graphs from various institutions would be merged.

4. Linking out beyond the Darwin Core data domain

Darwin-SW facilitates the linkage of core museum resources (specimens, taxa) to related, 'external' RDF resources developed in other knowledge domains (such as genes, locations, publications, agents, media, etc.). In particular, the existence of an Organism class permits simple linkage of taxonomic determination, vouchering information and *observations* of organisms. These observation data come from a wide range of biological disciplines, and include records of experimental treatments on individuals, repeated ecological measures of individuals (e.g., the many datasets for trees in ecological plots), and the observations of morphological details of the organisms from which specimens are derived.

As an example, Fig. 3 shows a graph representation of a morphological observation of a named Organism. It employs two additional, existing ontologies: the OBOE observations ontology[37] that instantiates the observation (which we treat as a `dsw:Token`), and OBO (Open Biological Ontologies[38]) whose terms to model the part of the organism observed. We model phenotype following the widely used EQ ('Entity-Quality') approach (e.g. [6]), where an instance of an anatomical class (for example from the Plant Ontology[39]), which is *part of* the Organism, is the *bearer of* a quality (from the PATO ontology[40]). If the observation is quantitative, the `oboe:Measurement` class and associated properties (dashed lines in Fig. 3) can be employed. Metadata about the observation (time, place) are represented in the Occurrence (see section 2.2.2). In this way, we can encode the morphological descriptions that are often part of the information on specimen labels (e.g., 'flowers blue').

Similarly, because Darwin-SW establishes a class for Organisms, it facilitates the documentation of interactions among organisms, such as predation, parasitism, and mutualism. Thus it also advances the ability to document ecological interactions using semantic tools.

[37] https://semtools.ecoinformatics.org/oboe
[38] http://www.obofoundry.org/
[39] http://www.plantontology.org/
[40] http://www.obofoundry.org/cgi-bin/detail.cgi?id=quality

5. The future of DSW

Because Darwin-SW is based on the Darwin Core standard, it should evolve over time as that standard evolves, with new terms being added or changed as necessary to maintain consistency.

Although changes to Darwin Core may require DSW to change, DSW was designed with stability in mind. The DSW URIs are based on a `purl.org` namespace so that they will remain independent of any particular server domain. As it becomes necessary to deprecate DSW terms, they will be maintained in the vocabulary with their deprecation noted in the RDF.

Advancing the efforts towards expressing biodiversity data as RDF depends critically on the availability of object properties to link resources described using Darwin Core terms. In this paper we have provided concrete examples of how Darwin-SW object properties can be used to accomplish reasoning tasks in the context of integration of biodiversity instance data. A variety of approaches to creating object properties have been suggested, and achieving a consensus on which approaches are effective requires testing whether those properties can be used to satisfy important use-cases using real data on a realistic scale. Using the examples in this paper, the performance of Darwin-SW in materializing useful entailments can be compared to other approaches. This is an important step in the development of a consensus RDF model for the biodiversity informatics community.

Acknowledgements

Steve Baskauf's participation in the Semantics of Biodiversity Symposium at TDWG 2013 was supported by the Research Coordination Network for the Genomic Standards Consortium (RCN4GSC, NSF DBI-0840989) and the Scientific Observations Network (SONet, NSF #0753144, OCI-Interop).

Campbell Webb is grateful for research support by the National Science Foundation (DEB-1020868).

Appendix

Table 3

QName namespace prefixes used in this paper

vocabulary name	prefix	URI
Darwin Core (literal values)	dwc:	http://rs.tdwg.org/dwc/terms/
Darwin Core (IRI values)	dwciri:	http://rs.tdwg.org/dwc/iri/
Darwin-SW	dsw:	http://purl.org/dsw/
Dublin Core	dcterms:	http://purl.org/dc/terms/
Dublin Core Type Vocabulary	dcmitype:	http://purl.org/dc/dcmitype/
Friend of a Friend	foaf:	http://xmlns.com/foaf/0.1/
WGS84 Geo Positioning vocabulary	geo:	http://www.w3.org/2003/01/geo/wgs84_pos#
Extensible Ontology for Observations	oboe:	http://ecoinformatics.org/oboe/oboe.1.0/oboe-core.owl#
Phenotypic Quality Ontology	pato:	http://purl.org/obo/owl/PATO#
Plant Ontology	po:	http://owlfiles.plantontology.org/
Resource Description Framework	rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
RDF Schema	rdfs:	http://www.w3.org/2000/01/rdf-schema#
XML Schema	xsd:	http://www.w3.org/2001/XMLSchema#
Web Ontology Language	owl:	http://www.w3.org/2002/07/owl#

Refer to http://www.w3.org/TR/REC-rdf-syntax/ for a description of QNames.

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix dcmitype: <http://purl.org/dc/dcmitype/>.
@prefix dwc: <http://rs.tdwg.org/dwc/terms/>.
@prefix dsw: <http://purl.org/dsw/>.

<http://bioimages.vanderbilt.edu/uncg/39> a dwc:Organism;
    dsw:hasIdentification <http://bioimages.vanderbilt.edu/uncg/39#2010-11-19weakleya>;
    dsw:hasOccurrence <http://bioimages.vanderbilt.edu/kirchoff/em2557#occ>.
<http://bioimages.vanderbilt.edu/kirchoff/em2557#occ> a dwc:Occurrence;
    dsw:hasEvidence <http://bioimages.vanderbilt.edu/kirchoff/em2557>,
                    <http://bioimages.vanderbilt.edu/specimen/ncu592804>.
<http://bioimages.vanderbilt.edu/uncg/39#2010-11-19weakleya> a dwc:Identification;
    dwc:dateIdentified "2010-11-19"^^xsd:date;
    dsw:idBasedOn <http://bioimages.vanderbilt.edu/specimen/ncu592804>.
<http://bioimages.vanderbilt.edu/kirchoff/em2557> a dcmitype:StillImage, dsw:Token;
    dsw:derivedFrom <http://bioimages.vanderbilt.edu/uncg/39>;
    dcterms:created "2010-10-19"^^xsd:date.
<http://herbarium.unc.edu/image/089765> a dcmitype:StillImage, dsw:Token;
    dsw:derivedFrom <http://bioimages.vanderbilt.edu/specimen/ncu592804>;
    dcterms:creator <http://biocol.org/urn:lsid:biocol.org:col:15495>.
<http://bioimages.vanderbilt.edu/specimen/ncu592804> a dwc:PreservedSpecimen, dsw:Token;
    dsw:derivedFrom <http://bioimages.vanderbilt.edu/uncg/39>.
```

Fig. 4. RDF/Turtle serialization based on the graph in Fig. 2

Examples queries as described in Section 3.

Denormalization and inconsistent term use. Provider 1 uses the DSW terms `dsw:atEvent` and `dsw:locatedAt` as intended. It exposes the graph shown in Example 1, which assigns the time of the record to the `dwc:Event` instance.

Example 1

```
provider1:occ a dwc:Occurrence;
     dsw:atEvent provider1:event1.
provider1:event a dwc:Event;
     dwc:eventDate "1983-01-19"^^xsd:date;
     dsw:locatedAt provider1:location.
provider1:location a dcterms:Location;
     dwc:state "Ohio".
```

Because of the ability to record the time of each Occurrence at a greater precision, Provider 2 considers each Occurrence to be recorded at a separate Event. It prefers to link Occurrences directly to Locations and assign the time of the record to the Occurrence instance (in other words, to denormalize the model to eliminate Event). The provider thus inappropriately links Occurrences to Locations using `dsw:locatedAt` as shown in Example 2.

Example 2

```
provider2:occ a dwc:Occurrence;
     dwc:eventDate      "1983-01-19T09:23:47-
05:00"^^xsd:dateTime;
     dsw:locatedAt provider2:location.
provider2:location a dcterms:Location;
     dwc:state "Ohio".
```

Assume the graphs from the two providers were combined into a single graph. To attempt to discover all Occurrences that were recorded in the state of Ohio, one could perform Query 1.

Query 1

```
PREFIX dsw: <http://purl.org/dsw/>
PREFIX dwc: <http://rs.tdwg.org/dwc/terms/>
SELECT DISTINCT ?occurrence
WHERE {
 ?occurrence dsw:atEvent ?event.
 ?event dsw:locatedAt ?location.
 ?location dwc:state "Ohio".
 }
```

Query 1 would discover `provider1:occ` from the first provider, but would not discover `provider2:occ` from the second provider because of the lack of an Event instance and inconsistent use of `dsw:locatedAt`.

To use DSW consistently, the second provider could create a blank node representing the Event that is not explicitly present in its database (Example 3).

Example 3

```
  provider2:occ a dwc:Occurrence;
       dsw:atEvent _:bnode.
  _:bnode a dwc:Event;
       dwc:eventDate    "1983-01-19T09:23:47-
05:00"^^xsd:dateTime;
       dsw:locatedAt provider2:location.
  provider2:location a dcterms:Location;
       dwc:state "Ohio".
```

If the Example 3 graph were merged with the Example 1 graph, Query 1 would discover both Occurrence instances: `provider1:occ` and `provider2:occ`.

Detecting inconsistencies using ranges, domains, and disjoint classes in Darwin-SW

Running Query 2 on the incoming graph to be screened constructs a graph containing triples that consist of type statements that are entailed by DSW domain declarations.

Query 2

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-
schema#>
CONSTRUCT {?resource a ?class}
FROM <http://example.org/incoming-data.rdf>
WHERE {
 ?property rdfs:domain ?class.
 ?resource ?property ?object.
 MINUS {?resource a ?class.}
}
```

The MINUS filter (available in SPARQL1.1) removes solutions to the query that are already explicitly declared in the incoming graph. A similar query can be used to construct a graph containing triples that consist of undeclared type statements that are entailed by DSW range declarations.

The constructed graphs containing the type statements entailed by range and domain declarations is merged with the incoming graph and the existing multi-institution graph to create a new graph that includes all type declarations that are declared explicitly or entailed by use of the DSW vocabulary. Inconsistencies caused when resources are instances of two classes that are declared to be disjoint by DSW can be discovered in the merged graph using Query 3.

Query 3

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT DISTINCT ?resource ?class1 ?class2
FROM <http://ex.org/entailed-types.rdf>
FROM <http://ex.org/incoming-data.rdf>
FROM <http://ex.org/multi-inst-data.rdf>
WHERE  {
 ?resource a ?class1.
 ?resource a ?class2.
 ?class1 owl:disjointWith ?class2.
 }
```

References

[1] J. Wieczorek, D. Bloom, R. Guralnick, S. Blum, M. Döring, R. Giovanni, T. Robertson, D. Vieglais, Darwin Core: An Evolving Community-Developed Biodiversity Data Standard. *PLOS ONE* **7**(1):e29715, 2012. http://dx.doi.org/10.1371/journal.pone.0029715

[2] TDWG Vocabulary Management Task Group, Report of the TDWG Vocabulary Management Task Group (VoMaG). S. Baskauf, É. Ó Tuama, D. Endresen, and G. Hagedorn, eds, 2013. Available from http://www.gbif.org/resources/2246

[3] R. Walls, J. Deck, R. Guralnick, S. Baskauf, R. Beaman, S. Blum, S. Bowers, P. L. Buttigieg, N Davies, D. Endresen, M. A. Gandolfo, R. Hanner, A. Janning, L. Krishtalka, A. Matsungaga, P. Midford, N. Morrison, É. Ó Tuama, M. Schildauer, B. Smith, B. J. Stucky, A. Thomer, J. Wieczorek, J. Whitacre, J. Wooley, Semantics in Support of Biodiversity Knowledge Discovery: An Introduction to the Biological Collections Ontology and Related Ontologies. *PLOS ONE*, **9**(3):e89606, 2014. http://dx.doi.org/10.1371/journal.pone.0089606

[4] S. J. Baskauf, Organization of occurrence-related biodiversity resources based on the process of their creation and the role of individual organisms as resource relationship nodes. *Biodiversity Informatics* **7**:17-44, 2010. https://journals.ku.edu/index.php/jbi/article/view/3664

[5] N. Franz, R. K. Peet, and A. S. Weakley, On the Use of Taxonomic Concepts in Support of Biodiversity Research and Taxonomy, in: *The New Taxonomy*. Systematics Association Special Volume Series 74, pages 63-82. Taylor and Francis, Boca Raton, 2008.

[6] C. Mungall, G. Gkoutos, C. Smith, M. Haendel, S. Lewis, M. Ashburner, Integrating phenotype ontologies across multiple species. *Genome Biology* **11**:R2, 2010. http://dx.doi.org/10.1186/gb-2010-11-1-r2