Biodiversity Information Standards
(TDWG)

www.tdwg.org

TDWG GUID Applicability Statement

Date:
9-Sep-2010

Status:
TDWG Draft Standard

Permanent URL:
http://www.tdwg.org/standards/150

Task Group:
TDWG Globally Unique Identifiers Task Group (GUID)

Contributors:
Kevin Richards (Landcare Research)

Abstract:
This document

1. Provides guidance on how to use GUIDs (Globally Unique Identifiers) to meet specific requirements of the biodiversity information community;
2. Applies to GUIDs in general; there is, or will be, a separate document for the applicability of each specific GUID technology.

Legal Notice:

Disclaimer:

Table of Contents

Index of Recommendations

Motivation

The TDWG Globally Unique Identifiers Task Group (TDWG GUID) [1][2], after meeting twice in 2006, recommended the use of the Life Sciences Identifiers (LSID [3]) to uniquely identify shared data objects in the biodiversity domain.

Demonstration LSID providers and services now exist and the GUID technology has been tested. These useful test cases have resulted in a variety of issues that concern LSIDs. More detail can be found in the LSID Applicability Statement.

When the TDWG GUID subgroup first looked at GUID technologies, there were relatively simple requirements. Since that time, requirements have evolved within GUID technologies and interlinked data.

LSIDs were seen as the appropriate GUID technology because they were a specific and independent protocol that forced data providers to carefully consider the allocation of each identifier. This makes LSIDs harder to implement than simple URLs. LSID technology also does not work by default with some semantic web technologies such as Linked Data (which require HTTP resolvable URI GUIDs). HTTP resolvable URIs (Uniform Resource Identifier) are equivalent to URLs (Uniform Resource Locator), and can be resolved by any Internet tool, whereas the resolution of URNs (Uniform Resource Name) differ for each type of URN, and are therefore not resolvable using basic HTTP resolution. LSIDs themselves work well as GUIDs, and will also work with most semantic web technologies, so long as there is no need for HTTP resolution of those GUIDs (e.g. use of LSIDs in a standalone triple store scenario). There has also been a trend, in the TDWG community, towards semantic technologies and linked data, which as stated, requires URIs that are resolvable using HTTP resolution. Linked Data [9] is the practice of applying HTTP resolvable GUIDs to all Internet resources and linking to other Internet resources by referring to other resource URIs within the metadata of the subject resource. In theory, this approach allows all data on the web to be inter-connected and navigable using basic web technologies, such as the HTTP protocol.

General lack of consensus of preferred GUID technology has led to this review of all possible GUID technologies.

This applicability statement specifies the recommendations for GUIDs in general, and for use in the biodiversity informatics community. For the applicability of specific GUID technologies, see the related GUID standard documents.

Terminology and Definitions

This document follows the specification for an Applicability Statement as set out by TDWG. More information on the TDWG standards process, standards and categories of standards is available at http://www.tdwg.org/standards/status-and-categories/.

Throughout this document we use the term **object** to refer to an entity or information about it. We refer to the organizations that disseminate objects as **providers**.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [4].

Throughout the document we present each recommendation inside a box followed by the rationale behind the recommendation as in the example below.

R1. <Recommendation statement>

<Rationale for the recommendation.>

GUID Technical Considerations

Common GUID properties:

- **Identifiers** (GUIDs) should be ***referentially consistent*** and ***resolvable*** in order to support tests of uniqueness and the acquisition of associated metadata.
  - **Referential consistency**: The property that a GUID always refers to a specific object. All information associated with a GUID is about the same object. The properties of the object are subject to change, but once a GUID is assigned to one object, it **cannot** be reused to refer to a different object.
  - **Resolvable:** The GUID may be presented to a service that returns information related to the GUID and its referenced object. For example, an HTTP URI is resolvable by Internet browsers and other HTTP clients. However, a GUID does not necessarily have to be *self-resolving*; that is, the GUID may be treated separately from the mechanism through which it is resolved. For example, a UUID (Universally Unique Identifier) is not self-resolving; but it may still be resolvable through a resolution service operating through a standard protocol, such as HTTP.
- **Resolution services** need to be available (***permanent*** and ***robust***):
  - Resolution services should be provided for GUIDs that have been made publicly available. This ensures that the data for all GUIDs is retrievable at any time. Steps should be taken to ensure these resolution services are always available, including the situation where there are institutional changes surrounding the resolution services infrastructure (e.g. transfer of resolution service to a new institute that will continue to provide the service into the future).

The following are common issues that arise when considering GUID technologies:

- **Resolvability.** There is often a requirement from data consumers for a reliable and consistent way to resolve Identifiers using the HTTP protocol. The LSID specification does not specify a default HTTP resolution. This is discussed more in the LSID Applicability Statement.
- **Data vs metadata.** What information is data and what information is metadata?
- **Content type.** What format or schema should be used for serving the data and metadata (i.e., what format does the consuming system expect)? Should there be a default?
- **Permanence.** How to maintain data and the GUIDs that refer to that data indefinitely.
- **Publishing and consuming of GUIDs.** How do we make GUIDs available on the web, or find out about existing GUIDs?

Uniqueness and Resolution

The global uniqueness of an identifier is often confused with the issue of resolution of the identifier. These two attributes of GUIDs can be distinguished and discussed separately.

For example a Universally Unique Identifier (UUID) is a globally unique identifier, but there are no widely known and used protocols for resolving a UUID over the Internet (unlike HTTP URIs). This form of GUID is perfectly acceptable for uniquely identifying data objects within a dataset.

Some identifiers therefore provide uniqueness, but not resolvability.

GUID Technology

R1. A GUID technology **should** be chosen from the list of recommended GUID types.

There are a variety of GUID options available for assigning to data objects on the web. The following are GUIDs that have been investigated by TDWG and are considered suitable for use in the biodiversity domain.

- HTTP URI (this technology is used as a basis for some of the following options)
- LSID — Life Science Identifier. See the LSID Applicability Statement.
- DOI — Digital Object Identifier.
- PURL — Permanent URL.
- UUID — Universally Unique Identifier.
- Handle System.

There is often an acknowledged GUID technology within specific domains, for example DOIs are commonly used for publications. These defaults should be followed where appropriate.

The organisation (or perhaps "ontology") of some types of GUIDs can be summarised as follows:

- **URI**
  - **http: URI**
    - **PURL**
      - OCLC PURL (= http://purl.org/[id])
    - http:-proxied DOI (= http://dx.doi.org/[DOI name])
    - http:-proxied LSID (= http://lsid.tdwg.org/[lsid])
    - [other kind of http: URI]
  - **urn:** URI
    - **urn:lsid**: URI
    - [other kind of urn: URI]
  - [other kind of URI (e.g. mailto:, tag:, ...)]
- **Handle**
  - **DOI**
  - [other kind of handle]
- **UUID**
- [other kind of GUID]

The traditional or well accepted and used web technology is the HTTP URI. All GUIDs of this type are resolvable over the Internet using basic HTTP web resolution.

HTTP URI

A Uniform Resource Identifier (URI) consists of a string of characters used to identify or name a resource on the Internet. A URI scheme defines a specific syntax and associated protocols for a collection of URIs.

HTTP URI is a URI scheme whose identifiers are prefixed with "`http://`". An HTTP URI can be used to locate network resources via the HTTP protocol, and therefore supports the Linked Data practices well.

LSID

The LSID Applicability Statement provides specific recommendations for the use of LSIDs.

An LSID is a particular kind of actionable identifier, recommended for use by TDWG, which

- enables global uniqueness by including an Internet domain name, which is itself subject to rules and procedures ensuring uniqueness, and
- uses the domain name system to locate a resolution service that enables a user to find out more about the entity to which an LSID refers.

An LSID provides a means to identify and locate a piece of biological data and/or metadata on the web. For a more detailed description see the LSID Resolution Project Homepage (http://lsids.sourceforge.net).

By themselves, LSIDs do not meet the requirements of Linked Data because they are not HTTP URIs. Standard Linked Data clients will not be able to handle them. One solution to this problem is to represent LSIDs as HTTP URIs.

For example, the bioguid.info Web site provides LSID resolution proxy services. Appending the LSID "`urn:lsid:ipni.org:names:20012728-1:1.1`" to "`http://bioguid.info/`" yields the HTTP URI http://bioguid.info/urn:lsid:ipni.org:names:20012728-1:1.1. That URI, when presented to a Web browser, produces an HTML document containing the metadata of the referenced name object.

DOI

The Digital Object Identifier (DOI) System is a digitally managed system for persistent identification of entities. The term "DOI" is understood to mean "digital identifier of an object", rather than "identifier of a digital object". As well as identifying content items such as digital files and digital media manifestations of intellectual property, DOI names can also identify physical objects, performances and abstract works. For example, they can be used to identify: e-texts, images, audio or video items and software. DOI names can also be assigned to related entities in a content transaction (e.g. licenses, parties, etc.) The DOI name is the identifier string that specifies a unique object (the referent); the DOI System [http://www.doi.org/] is the functional deployment of DOI names as identifiers in computer sensible form through assignment, resolution, referent description, and administration.

DOI names resolve to data specified by the registrant, and use an extensible metadata model to associate descriptive and other elements of data with the DOI Name. The DOI System is an implementation of the Handle System and of the indecs Content Model [http://cordis.europa.eu/econtent/mmrcs/indecs.htm] and so inherits the design principles and features of each.

The DOI System is implemented through a federation of DOI Registration Agencies, under policies and common infrastructure provided by the International DOI Foundation, which developed and controls the system.

Major applications currently include persistent citation in scholarly materials (journal articles, books, and similar materials) through CrossRef [http://www.crossref.org/]; scientific data sets, through a consortium of leading research libraries and technical information providers, building on work by the German National Library of Science and Technology (TIB); and European Union (EU) official publications, through the EU publications office.

[Wikipedia: http://en.wikipedia.org/wiki/Digital_object_identifier]

PURL

A persistent uniform resource locator (PURL) is an HTTP Uniform Resource Identifier (URI) (i.e. location-based Uniform Resource Identifier or URI) with a redirect mechanism. It does not directly describe the location of the resource to be retrieved but instead describes an intermediate (more persistent) location which, when retrieved, results in redirection (e.g. via a 302 HTTP status code [http://www.w3c.org/Protocols/rfc2616/rfc2616-sec10.html]) to the current location of the final resource.

Persistence problems are caused by the practical impossibility of every user having their own domain name, and the inconvenience and money involved in re-registering domain names, that results in WWW authors putting their documents in arbitrary locations of questionable persistence (i.e. wherever they can get the WWW space).

[Wikipedia: http://en.wikipedia.org/wiki/PURL]

UUID

A UUID (Universally Unique Identifier) is a GUID created by an algorithm that virtually guarantees that no two identical UUIDs will ever be generated at any time or place. This avoids the need to check for identical GUID values when generating identifiers, and ensures that any search application will only return a single interpretation.

By themselves, UUIDs do not meet the requirements of Linked Data because they are not HTTP URIs, and indeed the identifier itself does not contain any information on how to resolve it. Linked Data clients, which are commonly only capable of resolving HTTP URIs, will not be able to handle them, nor will unqualified UUIDs be easily resolved (except, perhaps, through a web-wide service such as Google). One possible solution to this problem is to represent UUIDs as HTTP URIs.

For example, the zoobank.org web site provides UUID resolution proxy services. Appending the UUID "`8BDC0735-FEA4-4298-83FA-D04F67C3FBEC`" to "`http://zoobank.org/`" yields the HTTP URI http://zoobank.org/8BDC0735-FEA4-4298-83FA-D04F67C3FBEC. That URI, when presented to a Web browser, produces an HTML document containing the metadata of the referenced name object.

UUIDs may also be implemented as the `object identifier` portion of an LSID (see the LSID Applicability Statement), which itself may be resolved either directly through LSID resolution protocols, or by representing the LSID as an HTTP URI. For example, the UUID indicated above forms the object identifier portion of the LSID "`urn:lsid:zoobank.org:act:8BDC0735-FEA4-4298-83FA-D04F67C3FBEC`", which itself can be resolved by representing it as the HTTP URI http://zoobank.org/urn:lsid:zoobank.org:act:8BDC0735-FEA4-4298-83FA-D04F67C3FBEC.

Dissociating the identifier from the mechanism or protocol through which it is resolved has advantages and disadvantages, which will only be evaluated within the biodiversity informatics domain once systems that incorporate and resolve UUIDs are further developed.

[Wikipedia: http://en.wikipedia.org/wiki/Universally_Unique_Identifier]

Handle

The Handle System [http://www.handle.net/] is a technology specification for assigning, managing, and resolving persistent identifiers for digital objects and other resources on the Internet. The protocols specified enable a distributed computer system to store identifiers (names, or handles), of digital resources and resolve those handles into the information necessary to locate, access, and otherwise make use of the resources. That information can be changed as needed to reflect the current state and/or location of the identified resource without changing the handle.

The Domain Name System resolves domain names meaningful to humans into numerical IP addresses (locations of file servers). The Handle System is compatible with DNS but does not necessarily require it, unlike persistent identifiers such as PURLs or LSIDs which utilise domain names and are therefore ultimately constrained by them.

[Wikipedia: http://en.wikipedia.org/wiki/Handle_System]

HTTP GET Resolution

R2. HTTP GET resolution **must** be provided for non-self-resolving GUIDs.

For non-self-resolving GUIDs, such as UUIDs, resolution of that GUID via the HTTP protocol's GET method (the standard method by which a resource is retrieved on the web) must be implemented. This ensures that the data for the object being identified can be obtained from the provider of that GUID with tools that a majority of Internet users and developers already understand and use.

1. GUID Assignment

R3. Providers **must** assign at most one GUID to any particular object.

It makes sense that an identifier refers to one and only one object. However there has been a lack of clarity on what exactly an "object" refers to. It is possible for the identifier to refer to a physical object, an idea, a digital record, a set of records, and hence this point needs clarification. Specific issues arise when you have an object that does not exist as a physical object, such as a taxon concept. It makes sense to give physical objects, such as specimens, a single identifier that everyone reuses, but for abstract objects, this becomes harder to enforce when most data holders have their own versions of the abstract concepts.

The task of integrating all data holder concepts and identifiers is currently considered an impossible task. It therefore follows that separate identifiers will exist for possibly the "same" object. Following the linked data [9] model for globally integrated data, the preferred method to handle this situation is to ensure that links are maintained between the various versions of the same concept.

Examples of objects in the biodiversity domain that **should** be assigned GUIDs—

- scientific names;
- taxonomic concepts;
- taxon name usages;
- observations;
- individual organisms and observations;

- published and unpublished reference citations (e.g., literature);
- specimens;
- collections;
- images, videos, and sound recordings.

There are a variety of states of objects that can be assigned identifiers. The following describes different states, or types, of objects that can be described.

Physical objects

(*Sensu* Dublin Core class: `PhysicalResource`, e.g. a specimen.) A record for this type of object would contain metadata, but the physical artefact being identified would not be deliverable when the identifier was resolved.

Digital objects

(*Sensu* digital versions of Dublin Core classes: `StillImage`, `MovingImage`, `Sound`. There may be others that are not classed by Dublin Core.) In addition to the metadata that would be available for these objects, the actual data representing the object itself could be retrieved.

Observation objects

(No Dublin Core class.) These objects represent a particular defined occurrence, but refer to neither a physical artefact nor a retrievable data object. Measurements associated with the observation may be represented as metadata. Observation objects should not be typed as events because an observation object is created as the result of an event (as is the case with a physical or digital object) but it does not represent the event itself.

Abstract objects

These objects represent concepts. They have neither physical nor electronic representations, but are distinguished from observation objects in that abstract objects are subject to differing definition, while observation objects are not (they tend to be more fact based). Separate identifiers may exist for abstract objects (and hence need to be related as was described under recommendation 3), but should not for observation objects. An example of an abstract object is a Taxon Name, as opposed to a physical object such as a specimen.

R4. Only one globally unique identifier should be assigned to each object.

Wherever possible, a data provider should assign a single GUID as the identifier of an object. Assigning more than one GUID to a single object is counter-productive because it—

- makes it more difficult for clients to check object identity and detect duplicates;
- increases the cost of maintaining the identifiers (e.g., more records are needed, more effort is required to prevent and correct assignment errors).

This does not exclude the use of multiple GUID technologies for a single object. However, a consistent approach to defining these GUIDs should be used. For example you may have a non-self-resolvable GUID, such as a UUID, and then render it as a self-resolving GUID by embedding it within a standard resolution protocol. For example:

GUID (a UUID):
    `8A5D181B-88F6-47AE-B310-2BED677C73D2`

LSID:
    `urn:lsid:example.org:stuff:8A5D181B-88F6-47AE-B310-2BED677C73D2`

PURL:
    `http://purl.org/example/stuff/8A5D181B-88F6-47AE-B310-2BED677C73D2`

This approach also improves the separation of Identifier and Resolution Technology, where in the above example, the Identifier component is the UUID and the Resolution Technologies are LSID, and PURL.

If multiple GUIDs are used, then an effort should be made to link the GUIDs through the metadata that is provided for each GUID.

R5. Providers should only assign GUIDs to objects for which they are the authority.

By assigning a GUID to an object, a provider is stating that it is responsible for it. Clients are able to retrieve attribution information about the object by resolving its GUID. This creates a strong bond between the object and its provider.

However, in many cases there will be no official authority for the data in question—for example a taxon name—and different identifiers will be assigned to the same object by a number of providers. In this case it is important to reuse existing identifiers wherever possible, and also to define two instances of the same object as equal using constructs such as `owl:sameAs`.

Providers should express object attribution in an appropriate manner according to the standards for the particular resource type, e.g. the Dublin Core [5] metadata term **`creator`**.

R6. Aggregators should assign new GUIDs to derived objects.

Aggregators add value by collecting and integrating data from distributed, heterogeneous sources. Added value may come from:

- Integrating objects into homogeneous datasets;
- Verifying consistency;
- Georeferencing locality descriptions;
- Checking spelling or
- Resolving ambiguities

Aggregators then serve the value-added objects to clients and become the authority for the modifications made to the original objects. Aggregators **should** assign GUIDs to value-added objects they create. They should also reference the aggregated objects (see recommendation 9). However, aggregators should reuse the existing GUIDs if they have not modified the original object (see Recommendation 8).

R7. GUIDs should be resolvable.

GUIDs are not guaranteed to be resolvable. To attain all benefits of GUIDs however, such as source attribution, providers of GUIDs **should** provide resolution of their GUIDs indefinitely. This is independent of whether information concerning the resolution mechanism is part of the GUID itself (e.g., HTTP URI and LSID), or appended to the GUID (e.g., a DOI or a UUID resolved through an appropriate resolution mechanism, such as HTTP or LSID.

2. Reuse of Existing GUIDs

R8. Information systems should use existing GUIDs when available to refer to external objects.

Information systems keep relationships between objects from different sources. The reference to the original object is usually lost or weakened due to the lack of a standard object identifier. To ensure the original data is accessible, either the original GUID should be used, or a link should be provided back to the original data.

R9. Aggregators should use GUIDs and the Dublin Core metadata term `source` to link derived objects to their sources.

Using GUIDs to link value-added objects to their sources is important to—

- Give the value-added object proper attribution. Clients may use the GUID of the source to retrieve information about the original object and its creator.
- Allow clients to detect duplicates of the original object that may have been modified and served by different providers.

The Dublin Core term **`source`** is appropriate to express this relationship because it has the appropriate meaning and is part of the most popular metadata vocabulary available. It may be more suitable to use more specific relationship predicates in a particular case, for example the Darwin Core [8] term, `relatedResourceID`.

3. GUID Data and Metadata

The specific format of the data and metadata that a GUID resolves to will vary depending on factors such as the object type, and the scope of the data source.

There are several recommendations that are encouraged for consistency within the biodiversity domain.

R10. The default metadata response format **should** be RDF serialized as XML.

If no format is specified in a resolution request, then GUID authorities should return information in RDF format by default. Other formats may be returned if supported by the provider. This ensures the default metadata for an object is compatible with semantic web technologies and can be used for semantic analysis and inference.

R11. Objects in the biodiversity domain that are identified by a GUID should be typed using the TDWG ontology or other well-known vocabularies in accordance with the TDWG common architecture.

Any objects identified by a GUID in the biodiversity domain should be typed or classified using the TDWG ontology vocabularies [6] or other well-known vocabularies. The type of the object can also be thought of as the basis of the digital record, such as a "PreservedSpecimen". Typing must follow TDWG common development architecture [7]. If standard ontologies already exist, they should be used or extended where necessary instead of adding new, custom-built ontologies.

Machine and human clients that retrieve the metadata associated with a GUID will use the associated classification (type) information to decide how to process the metadata and any associated data. If the classification information is novel, processing may be difficult or impossible. Use of well known classes (types) allows the development and integration of applications that exploit the known classes.

Examples of ontology domains suitable for biodiversity objects include:

- General metadata; (e.g. Dublin Core, http://dublincore.org/; Darwin Core, http://rs.tdwg.org/dwc/)
- Taxon name / concept; (e.g. TCS, http://www.tdwg.org/standards/117/)
- Literature;
- Specimens / Occurrences; (e.g. ABCD, http://www.tdwg.org/standards/115/; Darwin Core http://rs.tdwg.org/dwc/)
- Collection metadata; (e.g. NCD, http://www.tdwg.org/standards/312/ [draft])
- GIS data. (e.g. GML, http://www.opengeospatial.org/standards/gml)

The granularity of the data for specific object types is also an issue. For example, when resolving a GUID for a specimen, should the provider supply pure specimen data such as the Accession Number and Collector Name, or should more extended information be provided such as Identifications, Taxon Concepts and Names of those Identifications, Locality, and Person/Collector details? This is a task that needs to be considered depending on use cases, the type of data and the type of data source.

The following are general guidelines for making these decisions:

- What is considered to be the primary object?
- What type of objects would you consider applying GUIDs to? What type of objects would consumers/users be likely to request? For example, a herbarium is likely to want to provide specimen data, and it would be reasonable to expect identification and gathering data to be provided with a specimen, and perhaps taxon concept data. It is likely however that consumers will request details for a specific taxon concept separately so the main objects could be the specimen and the taxon concept. GUIDs could be applied accordingly, and the relevant scope data would be supplied by the provider for those GUIDs.

The degree of granularity of the data provided will determine how efficiently users are able to query a data source and obtain precise and accurate answers to their requests.

4. GUID Versioning

R12. Working Groups and Use Case Engineers should determine the degree of change required for GUID reassignment.

For each use case and application domain for GUIDs there will be a degree of change that will define when the associated object has changed enough to fundamentally alter the meaning of that object. This could be done by listing the properties of the data object that are considered “core” to that object, and what degree of change to those properties will result in a fundamental change, and hence require a new GUID to be assigned.

R13. GUID authorities should use appropriate metadata properties to represent relationships between revisions of an object.

GUID authorities should use the following Dublin Core RDF and OWL properties to represent relationships between revisions of an object:

- **`dcterms:replaces`** — Points to the revision superseded by the revision at hand.
- **`dcterms:isReplacedBy`** — Points to a newer revision that supersedes the revision at hand.
- **`dcterms:hasVersion`** — Links an object to its revisions, regardless of whether it supersedes or is superseded by the other revisions.
- **`owl:versionInfo`** — String with information about the revision, such as the LSID revision identification and revision control keywords.

R14. Clients must not try to infer relationships between objects based on any part of a GUID. Instead, clients must dereference the GUID and retrieve any assertions about revisions from the returned metadata.

Clients may be tempted to infer relationships between objects associated with revision enabled GUIDs, such as LSIDs that differ only on the revision identifier. For example, the 2 LSIDs `urn:lsid:example.org:name:12345:a` and `urn:lsid:example.org:name:12345:b` only differ by the version of the LSID (i.e., “a” and “b”). Consumers of these IDs may decide to make an assumption that the IDs are the same because they are not concerned about small version differences. This practice is not encouraged because the semantics of revision identifiers is not defined in the LSID specification. See the LSID Applicability Statement for more information.

References

[1] TDWG Globally Unique Identifiers Task Group (TDWG GUID) Website: www.tdwg.org/activities/guid/ (accessed 27-Aug-2007)

[2] TDWG Globally Unique Identifiers Task Group (TDWG GUID) Wiki: http://wiki.tdwg.org/GUID/ (accessed 27-Aug-2007)

[3] Life Sciences Identifiers Specification. OMG Specification, 2004: http://www.omg.org/cgi-bin/doc?dtc/04-05-01 (accessed 27-Aug-2007)

[4] Internet Engineering Task Force (IETF). RFC 2119. Key words for use in RFCs to Indicate Requirement Levels. http://www.ietf.org/rfc/rfc2119.txt (accessed 17-Dec-2010)

[5] Dublin Core Metadata Initiative (DCMI): http://dublincore.org/ (accessed 27-Aug-2007)

[6] TDWG Ontology: http://wiki.tdwg.org/twiki/bin/view/TAG/LsidVocs (accessed 27-Aug-2007)

[7] TDWG Technical Architecture Interest Group (TDWG TAG) Website: http://www.tdwg.org/activities/tag/ (accessed 27-Aug-2007)

[8] Darwin Core Standard: http://rs.tdwg.org/dwc/terms/ (accessed 14-Jan-2011)

[9] Linked Data. http://linkeddata.org/