

Recommendations for implementation of guides in the SERNEC collections community

Steve Baskauf, Vanderbilt Univ. Dept. of Biological Sciences, steve.baskauf@vanderbilt.edu
Ver. 1.1, (2010-05-20). For the most recent version, go to <http://bioimages.vanderbilt.edu/guid>

Contents

I. Background on globally unique identifier (guids) in the context of Biodiversity Informatics.....	3
A. Background.	3
B. Three Requirements for valid guides (and one suggestion).....	4
C. Types of guides and why we need them	5
The Web of information.....	5
What about LSIDs?.....	5
HTTP URIs as guides	6
D. A rule about HTTP URI guides: different representations must have different URIs... 	7
E. Summary	8
II. Guid implementation issues	9
A. Barriers to the implementation of guides	9
B. Relationship between IT capabilities and guid requirements.....	10
Categories of institutions based on their IT limitations.....	10
Relationship of the "persistence" requirement of guides to IT infrastructure	11
C. Overcoming Barrier 1: Creating a guid format.....	11
1. Acquisition of a stable domain or subdomain.....	11
2. Use of a "namespace" as a part of your localIdentifier.....	12
3. Caveats.....	12
4. Summary	14
D. Overcoming Barrier 2: Determining the metadata that should be provided for particular types of resources and the terms that should be used to describe them	14
1. Conceptualizing metadata terms.....	14
2. Summary of the functions of metadata terms	16
3. How do we know which metadata terms we actually need?	17
4. Terms which should probably be databased	17
5. Terms which should be exposed in the RDF (and not already on the previous list).	19
E. Overcoming Barrier 3: Determining the format of the RDF files associated with the guid.....	19
1. Conflict of interest between live plant photographers and specimen databasers.....	19
2. Default guides for source Individuals and images of specimens when not explicitly assigned.....	21
3. Rule for constructing HTTP URIs for determinations of an individual or specimen when one is not explicitly assigned	22
5. Conceptual representation of images.....	24
6. Strategy for simultaneously accommodating simple and complex models for occurrence metadata relationships.....	26
7. General structure of RDF files when Individuals are assigned URIs independently from their associated Occurrences (images and specimens)	26

8. General structure of RDF files for specimens that have a single image.....	28
9. Advantage of this approach #1: Ability of "foreign" authorities to link to guids.....	32
10. Advantage of this approach #2: Linking duplicate specimens	34
F. Overcoming Barrier 4: Figuring out how to implement the delivery of the HTML and RDF files	35
1. RESTful services.	35
2. Representations.	36
3. Methods of file serving.	37
4. Long-term flexibility for the data provider.	38
5. Relationship between HTTP URI format and maintaining flexibility of file generation method and method of providing representation.	38
6. Relationship between institution type and file generation method.	39
7. Technical details of delivery option 1: Redirection to an HTML file by URL rewrite and access to RDF files by the link method.	39
7. Technical details of delivery option 2: Content negotiation to static RDF and HTML files having the same base URI as the GUID.	41
8. Technical details of delivery option 3: Accomplishing content negotiation and file generation dynamically using generic programmable server software.....	43
9. How do we get there from here?.....	45
III. How will a Linked Data system of biodiversity resources identified by guids be used to do anything useful?	46
A. The problem of the chicken and the egg.	46
B. How and why do you assemble a database from records that are scattered across the planet?.....	46
C. Aside from altruism, is there any benefit for me to do this?.....	47
D. Conclusions.....	48
 Appendix A - Reference resources	 49
Appendix B - RDF example for a specimen, its individual, and its image in a single file.....	51
Appendix C - RDF example of metadata for an individual in a separate file.....	54
Appendix D - RDF example of metadata for a live plant image in a separate file.....	57

Note: the following namespace abbreviations are used in this document.

rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 rdfs="http://www.w3.org/2000/01/rdf-schema#"
 dcterms="http://purl.org/dc/terms/"
 dwc="http://rs.tdwg.org/dwc/terms/"
 owl="http://www.w3.org/2002/07/owl#"
 sernec="http://bioimages.vanderbilt.edu/rdf/terms#"
 xmp="http://ns.adobe.com/xap/1.0/"
 xmpRights="http://ns.adobe.com/xap/1.0/rights/"
 Iptc4xmpExt="http://iptc.org/std/Iptc4xmpExt/2008-02-29/"
 mbank="http://www.morphbank.net/schema/morphbank#"
 mix="http://www.loc.gov/mix/v20"
 bibo="http://purl.org/ontology/bibo/"
 foaf="http://xmlns.com/foaf/0.1/"
 stdview="http://bioimages.vanderbilt.edu/rdf/stdview#"

At this time, a namespace has not been declared for mrtg: . So at this point mrtg terms can't be resolved as RDF.

I. Background on globally unique identifier (guids) in the context of Biodiversity Informatics

Note: this document represents the opinion of Steve Baskauf and is not an official policy of the Southeastern Network of Expertise and Collections (SERNEC), Vanderbilt University, or any other organization. Hopefully this document will change and improve through feedback as we better learn to use guid. I also apologize in advance for errors caused by my general ignorance. Please send comments, corrections, and feedback to steve.baskauf@vanderbilt.edu.

[Some of this material in this section is recycled from my 2010-05-09 report to the SERNEC Live Plant Imaging Group.]

A. Background.

The following material (supplemented by miscellaneous web pages) forms the basis for this section and is recommended reading for those who want to better understand the technical details of guid. Additional resources are listed in Appendix A.

[Cool URIs for the Semantic Web \(http://www.w3.org/TR/cooluris/\)](http://www.w3.org/TR/cooluris/) [background on HTTP URIs]

[Adoption of Persistent Identifiers for Biodiversity Informatics - Recommendations of the GBIF LSID GUID Task Group - 6 November 2009 \(http://www2.gbif.org/Persistent-Identifiers.pdf\)](http://www2.gbif.org/Persistent-Identifiers.pdf)

[TDWG GUID Applicability Statement, draft standard, 3-Sep-2009 \(http://www.tdwg.org/stdtrack/article/download/150/51\)](http://www.tdwg.org/stdtrack/article/download/150/51)

[How to Publish Linked Data on the Web \(http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/\)](http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/) [background on the Linked Data concept]

The TDWG document was submitted as a standard last fall and had a brief comment period, which I took as an indication that it was on the fast track to adoption. However, many months have now passed with no information about the status of the standards adoption process.

There are several terms used for identifiers in the context of biodiversity informatics. In addition to "globally unique identifiers", these include "persistent actionable identifiers", "persistent identifiers", and "universally unique identifiers". The distinction among these terms is described in section 2.2 of the GBIF document. For convenience, I will use "guid" to refer to identifiers that have the characteristics described in the following sections. Another term that will be used repeatedly in this paper is "resource". A resource can essentially be anything that can be assigned an identifier. That would include information resources such as a web pages and images, but could also include physical resources (e.g. specimens, people) and conceptual resources like species concepts.

B. Three Requirements for valid guides (and one suggestion).

Globally unique. The most obvious requirement for guides is that they be globally unique, i.e. guaranteed to be different from any other identifier on the planet. In the consensus that has emerged in the biodiversity community, there is one (and only one) accepted way to accomplish this: using an Internet domain (or subdomain) name. A domain name is the part of a URL that comes after the "http://", such as vanderbilt.edu or ser nec.org . Domain names are allocated to institutions or "bought" from a broker like godaddy.com. A subdomain is a modification of a domain name created by adding other characters in front of the domain name and separating the additional parts by using dots. For example bioimages.vanderbilt.edu is a subdomain of vanderbilt.edu and www.ser nec.org is a subdomain of ser nec.org . The "owner" of the domain (or subdomain) is the person or institution that controls what is made available on the Internet under that domain.

In the past, there was some discussion of creating guides using the Darwin Core "triples" of institutionCode:collectionCode:catalogNumber . Forget about this - it isn't guaranteed to be unique and it won't work.

All versions of guides acceptable to the Biodiversity informatics community depend on merging a local identifier with a domain (or subdomain) name. It is the responsibility of domain owners to make sure that their local identifiers are unique within their institution. If that is done, the guide as a whole is guaranteed to be unique since the rules of the Internet prevent any two people or organizations to control the same domain name.

Actionable. The second requirement of guides is that they are actionable. That means that it must be possible to use the guide to find out information about the resource that is being identified. For humans, this is pretty simple - the guide should produce a web page with information about the resource (metadata) when the guide is typed into a web browser. The trickier aspect of this is that a computer (or more accurately a computer program designed to search the web for information) must also be able to get metadata information about the resource when it obtains the guide. The form of this information is called resource description framework (RDF) and the information is provided in a file using XML format rather than the HTML format used by web pages.

Persistent. The third requirement for guides is that they are persistent. In layman's terms, that means that they should stay the same and continue to provide metadata forever. Given that the Web hasn't been around very long, forever is a pretty long time. But at a minimum, guides should stay the same for a long time (think of how often a bookmarked link has disappeared on you).

There are two practical implications of the requirement for persistence. One is that a guide should never be created using a domain name that is temporary. Domains (or subdomains of domains) that are owned by institutions are good. Domains purchased from GoDaddy on sale for \$.99 are bad. The second implication is that the domain owner should plan for guide names that will not need to be changed. So guides that depend on a particular kind of scripting language or some a particular web server name (either of which might change in a year) are bad. Example:

<http://freds-server.myorg.com/12345.php>
is a bad guid
<http://images.herbarium.org/12345>
is a good guid

Cool. A fourth recommendation (a suggestion but not a requirement) is that guides be "cool" (see "Cool URIs don't change" <http://www.w3.org/Provider/Style/URI>). In addition to persistence, cool guides are relatively short, sensible, and uncomplicated. For example, http://myuniversity.edu/fp339~8fe88.ax1199gG0OoQ12Iil-_.:345abz might be globally unique, actionable, and persistent, but it is horribly complicated and virtually impossible to type in a web browser (and therefore not "cool").

C. Types of guides and why we need them

The Web of information

The power of the Internet is that creating useful webs of information does not depend on a single person or institution. The World Wide Web works because people around the planet follow a few simple rules that allow users to find and understand content. Web browsers know how to display content to humans in useful ways because all content is written following the rules of some version of hypertext markup language (HTML). Other content can be found through links because the links refer to URLs (uniform resource locators) that follow naming rules that are a part of the hypertext transfer protocol (HTTP).

The Semantic Web extends the concept of the World Wide Web by linking not only web pages but many other kinds of data using additional sets of rules. This concept is called "Linked Data" (<http://linkeddata.org/>). The common language of the Semantic Web is resource description framework (RDF) which depends on a common set of terms to describe things (i.e. the metadata terms that we are trying to standardize). One data resource is linked to another through guides. Thus guides are the "glue" that holds the Semantic Web together and allows it to be built by many people in a distributed way rather than depending on a single institution to do all of the work.

What about LSIDs?

In recent years, considerable effort seems to have been exerted toward experimenting with Life Science Identifiers (LSIDs) and working out the technical details needed to make them work. In the process, the Biodiversity informatics community seems to have become greatly divided, with about half of the people loving them and half hating them. This division seems to have slowed down progress on guides considerably.

In the end, two things seem to have determined the fate of LSIDs: the technical problems in running an "LSID resolution service" seems to have been beyond the realm of mere mortals, and a developing consensus that guides should meet the requirements of "Linked Data", namely that an identifier must do something (i.e. be "resolvable") when you put it in a regular web browser. The consensus that came out in the TDWG and GBIF documents was that if LSIDs were to be used, they would also have to be resolvable in a "proxy form". In essence, a domain name beginning with "http://" would have to be strapped onto the front of them to make them usable.

At this point, there are very few sites that have actually implemented functioning LSIDs. One is Biodiversity Collections Index. You can see how LSIDs work there by going to <http://www.biodiversitycollectionsindex.org/>. The LSID for Bioimages is urn:lsid:biocol.org:col:35115 . If you put it in a web browser, nothing will happen. The proxy version of the LSID is <http://biocol.org/urn:lsid:biocol.org:col:35115> . If you put it in a web browser, you will get human-readable information about the Bioimages collection.

In the end, it appears to me that to mollify the supporters of LSIDs, LSIDs are considered an acceptable form of guid as long as they can be supported in their proxy form. However, an LSID in its proxy form is actually a form of a much simpler type of guid called an HTTP URI guid (see next section). So it seems to me rather pointless to go to the bother of trying to implement a more complicated system (LSIDs) when a more easily usable system (generic HTTP URIs) can be used instead.

One final note on LSIDs. Specify 6 claims to support LSIDs. This is not true. It WILL generate LSIDs, but using an incorrect format (containing 7 parts separated by colons rather than the maximum 6) and it will not (as far as I can see) provide any of the tools or metadata files needed to make the LSIDs actually "work". Compare the description of LSIDs in the Specify Help with "LSID naming conventions" at <http://www.ibm.com/developerworks/opensource/library/os-lsidbp/>

HTTP URIs as guides

There now seems to be a general consensus that guides should follow the practices of Linked Data. In particular, guides should be of a particular type known as an HTTP URI. The "HTTP" part means that the guid starts with "http://", which also means that it can be expected to produce something when typed in a web browser. The "URI" part means that the guid is a uniform resource identifier (URI). Most people are familiar with a type of URI known as a URL (uniform resource locator; a subset of URIs). A URL is both an identifier and a locator, because it uniquely identifies a web page and also can be used to request that the web page be sent to your web browser. However, a URI can identify something without delivering it to a web browser. For example

<http://bioimages.vanderbilt.edu/ind-baskauf/66920>

is an HTTP URI that identifies a particular valley oak tree in Mt. Diablo State Park in California. It is not reasonable to think that typing that URI into a web browser will cause the tree to be delivered to your computer through the Internet. Thus it is possible to create an HTTP URI that identifies an undeliverable resource (such as a specimen, species concept, or person) but that doesn't itself actually "do anything". However, there is a general expectation that URIs starting with "http://" will produce SOMETHING when you type them in a web browser, and according to the Linked Data philosophy and the rules for guides in Biodiversity informatics, they must. (The technical term for this is that URIs must be "dereferenceable". The act of returning data in response to a request for a server to dereference a URI is called "resolving" the URI.) In the case of the HTTP URI for the valley oak tree, typing the HTTP URI into your web browser actually produces an informational web page with the ugly looking URL

<http://bioimages.vanderbilt.edu/metadata.htm?baskauf/66921/metadata/img/3456/2304> .

This URL could change at any time while the URI that leads you to it will not. When a "linked data client" (computer program that can understand machine-readable RDF files) such as the OpenLink RDF Browser (<http://demo.openlinksw.com/rdfbrowser/>) requests information using the HTTP URI for the valley oak tree, it is sent the file <http://bioimages.vanderbilt.edu/ind-baskauf/66920.rdf>, an XML file containing metadata about the tree. The alternative forms of information that can be sent to a user in response to submitting the URI of a non-information resource are called "representations" of the resource.

So the HTTP URI <http://bioimages.vanderbilt.edu/ind-baskauf/66920> meets the requirements of a guid in that it is globally unique (the domain name "bioimages.vanderbilt.edu" is controlled by me and I have made sure that the local identifier "ind-baskauf/66920" is unique within the bioimages website), is actionable (a user receives an appropriate kind of file when the URI is resolved), and persistent (I don't intend for that URI to change or disappear at any time in the future). It is also relatively "cool" in that it is fairly simple and could easily be typed into a web browser.

D. A rule about HTTP URI guids: different representations must have different URIs

In contrast to LSIDs which are required to have a very specific format, there are relatively few rules about the constructing HTTP URIs. The most important one is that the URI of an abstract concept or physical object cannot be the same as the URI of the web page that represents it. In the valley oak example, the URI of the oak tree

<http://bioimages.vanderbilt.edu/ind-baskauf/66920>

is different from both the URI of the human readable web page

<http://bioimages.vanderbilt.edu/ind-baskauf/66920.htm>

(which forwards to the page with the ugly URL) and the machine readable RDF file

<http://bioimages.vanderbilt.edu/ind-baskauf/66920.rdf>

To make this method of distinguishing among resources work requires the web server to be able to perform **content negotiation** (redirecting users to different files depending on the kind of information they need).

Another typical way of making the URI of a non-deliverable object different from a file that represents it is by adding a "hash" ("#") character followed by some other characters (a "fragment identifier"). For example, the HTTP URI representing me (a thing not deliverable via the Internet)

<http://people.vanderbilt.edu/~steve.baskauf/foaf.rdf#me>

is not the same URI as the URL for the file that contains an RDF description of me

<http://people.vanderbilt.edu/~steve.baskauf/foaf.rdf>

(which is deliverable) even though the same thing happens if either of them is typed in a web or RDF browser. This is because servers ignore anything that comes after a "#" character. This method of distinguishing among resources that share the same root URI is called using a **hash URI**.

E. Summary

A consensus seems to have emerged about how guides should be created and behave.

1. Guides should be HTTP URIs.
2. Guides should be made unique by combining a domain (or subdomain) name with an identifier that is locally unique. Thus a guid should contain three parts:

[http://][domainName]/[localIdentifier]

e.g. [http://][bioimages.vanderbilt.edu]/[baskauf-ind/66920]

3. Guides should resolve to a web page when put in a web browser.
4. Guides should be linked to machine-readable RDF metadata in XML format.
5. One guid should refer to one resource (physical, conceptual, or information) and that guid should never change.
6. Different representations of a resource must have different HTTP URIs (e.g. a specimen cannot have the same URI as the web page about that specimen or an image of the specimen).

II. Guid implementation issues

A. Barriers to the implementation of guides

Now that there seems to be a consensus about the form that guides should take and what they should be able to do, why isn't everyone in the bioinformatics community using them? Aside from the obvious problem of lack of information about them, there are several other barriers to the implementation of guides:

- 1. Choosing a format for guides.** There is considerable flexibility in the form that HTTP URIs can take. However, because of the requirement that HTTP URIs be dereferenceable (return information when put into a browser), the format of the HTTP URI must be one that is compatible with the method that will be used to provide the information to users. The specific format of the RDF files associated with the guides will also affect the choice of format for URIs.
- 2. Determining the metadata that should be provided for particular types of resources and the terms that should be used to describe them.** Metadata are the data that describe the properties of a resource. Darwin Core, Dublin Core, and Media Resource Task Group (MRTG) are standard (or draft standard) lists of terms ("schemas") that should be used by the biodiversity informatics community to describe metadata. However, there are many more metadata terms in these schemas than any user is likely to ever want to use. A community of users should create a consensus of the core metadata terms that should be provided for particular resource types.
- 3. Determining the format of the RDF files associated with the guide.** The TDWG and GBIF documents specify that the RDF required to be associated with a guide be in XML format. This has several implications.
 - somebody has to understand how to create XML.
 - somebody has to understand how to create RDF in XML.
 - somebody needs to decide how the RDF will be structured in the files. Should related RDF be stored in a single file or several files? Should the RDF be broken into several XML container elements, each having its own guid or should all of the RDF be in a single container element? The answer to these questions is partly going to depend on the conceptual view of the identified resources and their interrelationships. The answer to these questions will, in turn, affect the format that works best for the HTTP URI guides.
- 4. Figuring out how to implement the delivery of the HTML and RDF files.** There are several options for how the required human-readable (HTML) and machine-readable (RDF) files can be associated with the HTTP URI of the resource and how those files will be generated and delivered by a webserver. The nature of an appropriate delivery system for a guide user will depend to a large extent on the user's access to IT resources and support, and the financial resources necessary to support those IT resources. That in turn will be influenced by the size of the institution. The chosen delivery system will also limit the possible format of the URIs.
- 5. A final challenge: Figuring out how a Linked Data system of biodiversity resources identified by guides will be used to do anything useful.** This isn't actually a barrier to the implementation of guides, but is an issue that should be considered before embarking on the course of implementing guides. How will a federated database be created using the RDF metadata made available through the resolution of HTTP URIs? If HTTP URIs never end up being used to aggregate data in a Linked Data sense, is there any point in going through the hassle of making guides capable of delivering RDF?

The overall implementation barrier caused by these individual challenges resembles a tangled thicket caused by the interrelatedness of these issues. Each barrier cannot be overcome individually without considering several of the others. Overcoming all of the barriers simultaneously requires an understanding of many issues and seems overwhelming. However, I believe that with careful consideration of the issues a solution is possible.

B. Relationship between IT capabilities and guid requirements

Categories of institutions based on their IT limitations.

Item 4 in the list above makes the point that the IT infrastructure available to an institution (i.e. herbarium, botanical garden, or museum) or individual will influence the methods by which the institution can meet the "actionable" (URI resolution) requirement of guides. For reference purposes, I will define four general categories of institutions based on the availability of IT resources to the institution.

Category 0. The institution (or individual) has few or no computing resources and no access to Internet services. The institution or individual is not a division of a larger organization that might be able to provide such services. Institutions in this category probably should not consider creating their own guides, but rather should seek to include their information resources (e.g. images) and metadata as a collection affiliated with another institution that has the resources to issue guides.

Category 1. The institution has access to computing resources and has created or plans to create an electronic database. This database could be as simple as an Excel spreadsheet or as complicated as software designed specifically to catalog collections (e.g. Specify). However, the database is on a stand-alone computer and is not part of a network. The institution has the ability to create a web presence but does not control its own web server. The institution does not have access to IT support personnel, or has IT support that does not have a vested interest in advancing the purposes of the institution.

Category 2. The institution has access to computing resources and has a database maintained by software specifically designed to handle collections. The database may be networked within the institution or beyond the institution via the Internet. The database can be exported using the built-in utilities of the software, but the institution does not have any means for customizing the software or creating software specifically designed for their institution. The institution has access to IT support personnel, either as a part of their own staff or through their institution, but does not have access to programmers. The institution has access to a web server and the ability to request changes to the server settings.

Category 3. The institution has access to computing resources and has a database maintained by software specifically designed to handle collections. The database may be networked within the institution or beyond the institution via the Internet. The institution has custom software to support their web presence and has access to programmers and server administrators who are

capable of integrating their collections database with their web presence. The institution controls their own server and has complete control over the server's settings.

Relationship of the "persistence" requirement of guides to IT infrastructure

Certain methods of implementing the requirements of HTTP URI guides are suitable for institutions in some categories, but not for institutions in another category. For example, institutions in Category 3 could fairly easily implement the generation of their HTML and RDF files dynamically, while this method would be completely impossible for institutions in Category 1. Category 1 institutions would need to create and deliver static files due to their restricted web capabilities. Maintaining static files would be managed relatively easily for a collection with a few thousand records, but would be virtually impossible for a collection containing hundreds of thousands or millions of records.

Certain HTTP URI guid formats would be suitable for a delivery system based on dynamic file generation, but not for static files and vice versa. One would hope that institutions would generally move up into higher categories as technology becomes less expensive and more widely available and as experience provides transferrable implementation templates that can be used by institutions that can't develop their own system. However, one should also consider that institutions might move down to lower categories if they lose funding or if a larger institution of which they are a part withdraws support for their division. The requirement that guides persist and remain unchanged over time dictates that it would be advisable for institutions to adopt a guid format that would work at any level rather than solely at the level into which the institution falls at the present.

These points will be discussed in more detail in section II.F.

C. Overcoming Barrier 1: Creating a guid format

1. Acquisition of a stable domain or subdomain.

Institutions having an IT infrastructure robust enough to support issuing their own guides (Categories 1-3) should probably acquire a stable domain/subdomain. If you are the owner of a domain that is likely to continue to function for decades, then you are ready to go. Unfortunately, few of us are in that position. `sernec.org` is a nice domain name, but when SERNEC's grant runs out in a few years it might disappear. The problem is that domain names that are long-lasting are likely to belong to big institutions and lowly photographers and herbarium curators are not likely to be in control of them. However, it should be relatively easy to obtain a subdomain for an institution with which you are affiliated. For example, my old website operated out of a subdirectory of the Vanderbilt College of Arts and Sciences web server (<http://www.cas.vanderbilt.edu/bioimages/>). This tied me to the `www.cas.vanderbilt.edu` subdomain over which I had no control. I now am using a subdomain of `vanderbilt.edu` (`bioimages.vanderbilt.edu`) over which I have complete control. If it became necessary to move my website and support of corresponding HTTP URI guides over to another web server at Vanderbilt, that would be no problem (although I couldn't move it to a different domain outside Vanderbilt).

So similarly, **herbarium.appstate.edu** or **herbarium.biology.appstate.edu** would be better than **www.biology.appstate.edu/herbarium** because what happens in that subdomain would be independent of decisions made by the biology department (which theoretically could be merged with another department or be eliminated at some point in the future). The same is true about **www.ulm.edu/herbarium** . In contrast, **tenn.bio.utk.edu** and **herbarium.bio.fsu.edu** are already suitable for use with HTTP URIs.

2. Use of a "namespace" as a part of your localIdentifier.

Creation of an identifier that is globally unique requires that owners of domain names are careful to use local identifiers that are unique within their domains. If you are assigning barcodes to your specimens or consecutive serial numbers to your images, you already have a unique identifier and you can simply append it to your domain name to create a guid. For example, let's say the domain of the herbarium is "herbarium.org" and that herbarium uses bar codes for specimens in the form "hb123456". In the herbarium's internal database, the number "123456" (or "hb123456") could be used as the institutions internal unique identifier. The HTTP URI guid for the specimen could be

`http://herbarium.org/123456`

or

`http://herbarium.org/hb123456`

This format has the advantage of extreme simplicity ("cool"; could easily be written down) and it produces a short identifier. It would be very suitable for a Category 3 institution having a webserver that generates files dynamically from its database. However, it would probably not be a good choice for a smaller institution with tens of thousands of records serving static files. If each record had a corresponding HTML and RDF file, a database having 10 000 records would have 20 000 files in the root directory of its website.

Another option is to create a locally unique identifier by concatenating a "namespace" with an object identifier (to use LSID terminology). The namespace might be a collection code (to use Darwin Core terminology) and the object identifier might be a catalog number. Another alternative would be to use a date (e.g. year) as the namespace and an accession number as the object identifier. Any combination of namespace and object identifier that makes sense in the context of a particular collection could be used. For example, since Bioimages has images from various photographers who may use the same image number, I ensure that image identifiers are unique by concatenating a unique namespace derived from the photographer's name (e.g. "baskauf") with an image identifier (such as the number assigned serially by my camera, e.g. "baskauf/66921"). There is also no rule in HTTP URIs that says that a local identifier can't be composed of three parts if that works for your system (e.g. "image/baskauf/66921").

3. Caveats

Reuse of object identifiers. One thing that you CANNOT do is to create a system that uses an identifier that might be reassigned to another object at a later time.

"Bad" characters.

Illegal: You cannot use the percent sign (%) except when "escaping" other characters (not advised). Don't use spaces, plus signs (+), asterisks (*), or exclamation marks (!). Characters outside of the Latin-1 character set can't be used.

Special use: Question marks (?) have a special meaning for identifying query strings and might be used after careful thought about how the URI will be resolved (probably safer not to use them). The hash (#) character has special use as a fragment identifier. Periods (.) may be used, but have a special meaning in that they imply a hierarchy. So don't use them without a well thought purpose.

XML problems: Because HTTP URIs will be used extensively in the XML RDF files associated with guides, the characters "<", ">", and "&" should not be used.

See http://www.w3.org/Addressing/URL/4_URI_Recommentations.html for more details.

Safe characters.

Latin-1 alphabetic characters (A-Z and a-z) are safe. Numerals are safe. Dash (-) and underscore (_) are safe for URIs and static file names.

The colon character is legal, but cannot be used in Windows file names. So it could be used in URIs by a Category 3 institution that uses dynamically generated files, but not by a Category 1 institution that stores static files on a hard drive, then uploads them via FTP to their file server.

The forward slash (/) character implies a hierarchical relationship among entities represented by the strings that it separates. It is fine for dynamic file generation or static sites that are organized hierarchically by directories (folders). (See section F.3. below for a discussion of static vs. dynamic file generation.)

The hash character should only be used specifically to create a fragment identifier in a hash URI.

Joining a namespace and object identifier to create a unique localIdentifier

There are several options for joining a namespace and an object identifier based on the "legal" characters discussed above:

colon	baskauf:66921
dash	baskauf-66921
underscore	baskauf_66921
nothing	baskauf66921
slash	baskauf/66921

The first example would not work for static files. The next three options would create a local identifier that was locally unique, but would not separate files into directories in a static system. The last option would work for either static or dynamic systems and is probably the best option.

Capital and lower-case letters

In theory, URIs containing strings that differ only in case should be interpreted as different identifiers (e.g. `http://url.com/hello.htm` vs. `http://url.com/Hello.htm`). As a practical matter, it is probably better to stick with URIs that contain only lower case letters.

This is conventional and avoids the problem of a user incorrectly writing down a URI of mixed case and then being unable to get anything when the URI is typed into the web browser.

4. Summary

The safest HTTP URIs probably:

- use subdomains of a large institution to ensure persistence with independence.
- contain only lower case letters, numbers, underscores (_), and dashes (-).
- indicate hierarchy using forward slashes (/).
- use periods only to separate parts of domain names and to separate file names and extensions.
- contain a localIdentifier consisting of a namespace and object identifier joined by a forward slash (exception: Category 3 institutions with stable dynamic website management can safely use a unique one-part localIdentifier)
- use hash URIs as necessary to differentiate non-information resources from the files that describe them

D. Overcoming Barrier 2: Determining the metadata that should be provided for particular types of resources and the terms that should be used to describe them

1. Conceptualizing metadata terms.

Most members of the collections community are familiar with the idea of metadata terms, particularly the use of Darwin Core terms to describe resource characteristics. Metadata terms pop up in many contexts and although the way they are displayed may change, their meaning is essentially the same: they describe properties of the object with which they are associated. I will illustrate some of these methods of representing metadata terms with an example. An herbarium specimen identified with the http URI <http://herbarium.org/hb123456> was collected by Jane Curator on June 23, 1997. Jane Curator has a FOAF ("Friend of a Friend") metadata file which describes her and has the URI <http://herbarium.org/people/jane-curator#person>. The Darwin Core metadata terms are prefixed by the namespace abbreviation "dwc:"

Database records

Here are the metadata represented as a table:

<i>occurrenceID</i>	<i>basisOfRecord</i>	<i>recordedBy</i>	<i>eventDate</i>
http://herbarium.org/hb123456	PreservedSpecimen	Jane Curator	1997-06-23

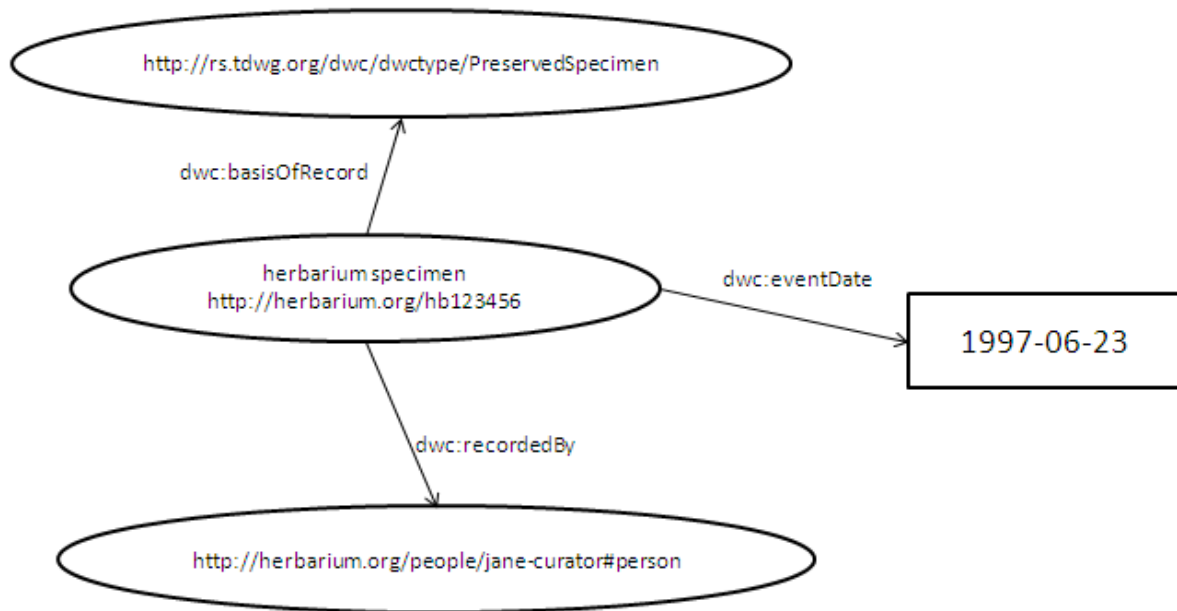
Here are the metadata represented in part of a generic XML file export from a database:

```
<occurrence>
  <dwc:occurrenceID>http://herbarium.org/hb123456</dwc:occurrenceID>
  <dwc:basisOfRecord>PreservedSpecimen</dwc:basisOfRecord>
  <dwc:recordedBy>Jane Curator</dwc:recordedBy>
  <dwc:eventDate>1997-06-23</dwc:eventDate>
</occurrence>
```

In both of these examples the metadata terms are used to identify elements of the database. The record in the table is represented by the table row and the individual database elements are the cells within the row. In the XML file, the record is delineated by the "container element" tag `<occurrence></occurrence>` and the individual elements are delineated by the various elements tagged by the term name (e.g. `<dwc:recordedBy></dwc:recordedBy>`). In both cases, the contents of the metadata elements are "strings" of characters (letters, numerals, and symbols).

RDF representations

Here are the metadata represented by an RDF graph:



In this example, the metadata terms describe the properties of the herbarium specimen. The arrows indicate relationships that can be described in "sentences" composed of a subject, predicate, and object (also known as an "RDF triple"). Three "sentences", each with the specimen as their subject, can describe the properties of the specimen (using abbreviated URIs):

```
hb123456 recordedBy jane-curator#person
hb123456 eventDate "1997-06-23"
hb123456 basisOfRecord PreservedSpecimen
```

Although these "sentences" are a bit odd sounding to humans, they actually can be understood by a computer program because the three predicates (*dwc:recordedBy*, *dwc:eventDate*, and *dwc:basisOfRecord*) are actually specified by HTTP URIs which refer to RDF descriptions of those Darwin Core terms. For example, *dwc:recordedBy* unabbreviated is the URI:

```
http://rs.tdwg.org/dwc/terms/recordedBy
```

Because the terms are identified by HTTP URIs, a computer can look them up via the Internet and "understand" (in a sense) what they mean.

The RDF graph format is really designed for humans to understand. The actual form in which the information is provided to a computer program is as an RDF/XML file. Here is part of an RDF file that would say the same thing as the graph above:

```
<rdf:RDF>
  <rdf:Description rdf:about="http://herbarium.org/hb123456">
    <dwc:basisOfRecord rdf:resource="http://rs.tdwg.org/dwc/dwctype/PreservedSpecimen"/>
    <dwc:recordedBy rdf:resource="http://herbarium.org/people/jane-curator#person"/>
    <dwc:eventDate>1997-06-23</dwc:eventDate>
  </rdf:Description>
</rdf:RDF>
```

Notice that the format of the XML file is similar to the example of the generic XML export file, but with several important differences. One is that the container element for the specimen record has an "about" attribute which tells a computer that the record is about a resource that is identified by the HTTP URI <http://herbarium.org/hb123456>. In first XML example, a computer would have no idea what the element

```
<dwc:occurrenceID>http://herbarium.org/hb123456</dwc:occurrenceID>
```

actually meant. The same thing is true about the *recordedBy* element. In generic XML export file, the element

```
<dwc:recordedBy>Jane Curator</dwc:recordedBy>
```

would mean nothing to a computer because "*Jane Curator*" is a literal string composed of symbols. In contrast, in the RDF file the element

```
<dwc:recordedBy rdf:resource="http://herbarium.org/people/jane-curator#person"/>
```

tells the computer that there is a resource elsewhere that describes the person who recorded the specimen. The computer could dereference that HTTP URI to find out about the person. The situation is the same with *dwc:basisOfRecord*. In the generic XML file, the value "*PreservedSpecimen*" is just a string of characters, while in the RDF file, the "resource" attribute points a computer to the place where it can find out what a *PreservedSpecimen* is. This ability to link resources is why RDF is the standard format for metadata in the Linked Data world.

2. Summary of the functions of metadata terms

From the examples above, you can see that metadata terms can serve several purposes:

- to act as "headers" or "tags" to identify elements in a database record
- to act as "predicates" of "sentences" that describe the properties of a resource

- to identify the relationship of the subject resource to other resources that are identified with HTTP URIs.

3. How do we know which metadata terms we actually need?

As you saw in the previous section, metadata terms can serve multiple purposes. Whether or not a particular term is "needed" depends on the situation in which it might be used. For example, does an herbarium specimen database "need" the term *dwc:basisOfRecord*? It would actually be a waste of time to include *dwc:basisOfRecord* as a field in every database record because it would have the same value (*PreservedSpecimen*) for every one. On the other hand, in an RDF file describing any of the specimens to the outside world, it would be very important to include the *dwc:basisOfRecord* element because that is the primary means of describing the type of thing that the resource represents.

Another circumstance in which a metadata term might not be needed in a local database, but be important when presenting metadata to the outside world would be terms that describe the relationship of one resource to another. For example, some variant of the specimen HTTP URI might be used to generate a hash URI for the specimen image (more on this in section II.E.2.) which could be a value for *dwc:associatedMedia* and *foaf:depiction* both of which relate the specimen resource to its image. Thus the generated image URI is not something that needs to be recorded in the local database under either of those two terms, but exposing values for those terms to the outside world would be important for explaining to the biodiversity community (with *dwc:associatedMedia*) and the rest of the world (with *foaf:depiction*) how the specimen and its image are related.

4. Terms which should probably be databased

For occurrences (specimens and images):

xmp:MetadataDate (i.e. the date when the metadata was last modified)

dwc:catalogNumber

dwc:collectionCode (if the catalog number is not already a locally unique identifier)

[Note: alternatively *individualID* (i.e. the HTTP URI guid) could be databased in lieu of generating it from catalogNumber or collectionCode+catalogNumber]

dwc:recordedBy (i.e. the collector)

dwc:eventDate (i.e. the date collected)

dwc:occurrenceRemarks

dwc:decimalLatitude *

dwc:decimalLongitude *

dwc:geodeticDatum (value is "epsg:4326" for GPS readings; "unknown" is an acceptable value)

*

dwc:coordinateUncertaintyInMeters *

dwc:locality

dwc:continent **

dwc:countryCode **

dwc:stateProvince **

dwc:county **

dwc:informationWithheld ***
dwc:dataGeneralizations ***

* The first three of these terms essentially form a globally unique identifier for location with the fourth specifying the uncertainty of location. These are the most important location terms for geolocated occurrences.

** These terms (along with *dwc:locality*) are the most important location terms for non-geolocated occurrences. In the case of geolocated occurrences, the values of these terms can be generated by software and would therefore not necessarily be databased.

*** For most occurrences, these terms will have null values. However, they should probably be included as database fields because they will be critical for protected species.

For determinations (i.e. identifications):

dwc:identifiedBy

dwc:dateIdentified

dwc:identificationRemarks

dwc:taxonConceptID (if the taxonomic data is to be looked up in another table)

[Note: if the taxon information is to be explicitly specified in the record rather than looked up, the hierarchical terms from the Darwin Core class *Taxon* should be used here. It is also possible that *dwc:taxonID* is a more appropriate term than *dwc:taxonConceptID* in some circumstances.]

For Individuals:

dwc:individualID

dwc:establishmentMeans *

* It is not clear whether this term should be associated with Individual source organisms or with the Occurrence records of that organism. For specimens that are the only record of occurrence for the Individual, this distinction is not very important.

Image-specific metadata:

dcterms:rights

xmpRights:owner *

xmpRights:UsageTerms (may be stored in the database in a more compact form, e.g. "BY-NC-SA") *

Iptc4xmpExt:CreditLine *

mrtg:caption

mbank:view

Iptc4xmpExt:CVterm **

xmp:Rating **

sernec:sernecImageCollectionStatus ***

dcterms:description ****

The following terms may be repeated for images having multiple Service Access Points (versions with differing quality, see http://www.keytonature.eu/wiki/Submission_v0.9#Service_Access_Point_Vocabulary and section E.5. below)

mrtg:variant

mrtg:providerManagedID (appropriate for storing the image file name for internal use)

dcterms:format (use MIME type, may be autogenerated if fixed to same type for all images)

mix:imageWidth (in pixels)

mix:imageHeight (in pixels)

mix:xSamplingFrequency (pixels per cm)

mix:ySamplingFrequency (likely to have the same value as *mix:xSamplingFrequency*; if so could be autogenerated)

* If all images in a collection have the same owner and usage terms, these would not need to be databased. Credit line may or may not need to be databased depending on whether the institution itself is to be credited or if multiple photographers need to be credited for their individual images.

** These terms are important for live plant images but not as relevant for specimen images.

*** If this term is not present, it would be assumed to have a value of zero.

**** This term would not need to be databased if it is generated automatically from other terms.

5. Terms which should be exposed in the RDF (and not already on the previous list).

However, they which would probably be generated based on an institutional default or which would have a fixed value for a particular type of resource and would therefore not need to be databased.

For occurrences (specimens and images):

mrtg:metadataLanguage (required by MRTG for images)

dcterms:type (required for MRTG for images)

dwc:collectionID (guid for collection, use Biodiversity Collections Index identifier)

dwc:institutionCode (standard for field, i.e. Index Herbariorum)

dcterms:creator (fixed at institution name for specimen images, same as *dwc:recordedBy* for occurrences derived from living organisms, see

<http://dublincore.org/documents/dcmi-terms/#terms-creator>)

dcterms:created (date photographed for specimen images, same as *dwc:eventDate* for occurrences derived from living organisms)

dwc:basisOfRecord

Image-specific metadata:

dcterms:title (required by MRTG; probably can be generated from *dcterms:description*)

xmpRights:WebStatement (probably generated from a more compact form of
xmpRights:UsageTerms)

mrtg:attributionLinkURL

mrtg:attributionLogoURL

The following terms may be repeated for images having multiple Service Access Points

mrtg:accessURL (may be autogenerated from internal *mrtg:providerManagedID*)

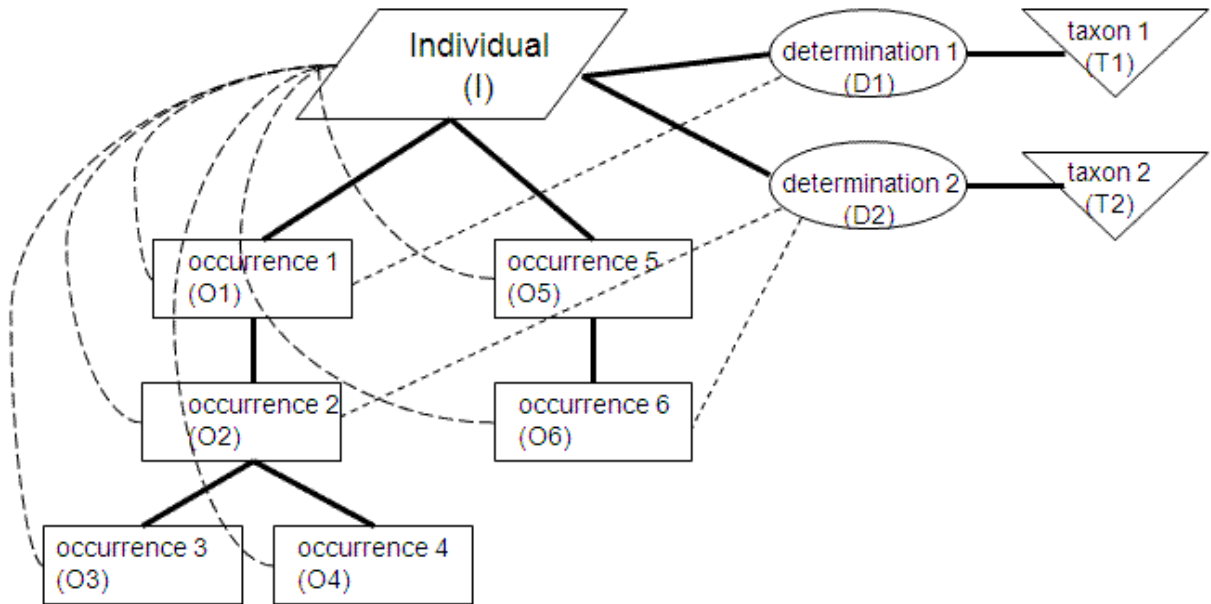
mix:samplingFrequencyUnit (fixed value of "cm")

E. Overcoming Barrier 3: Determining the format of the RDF files associated with the guid

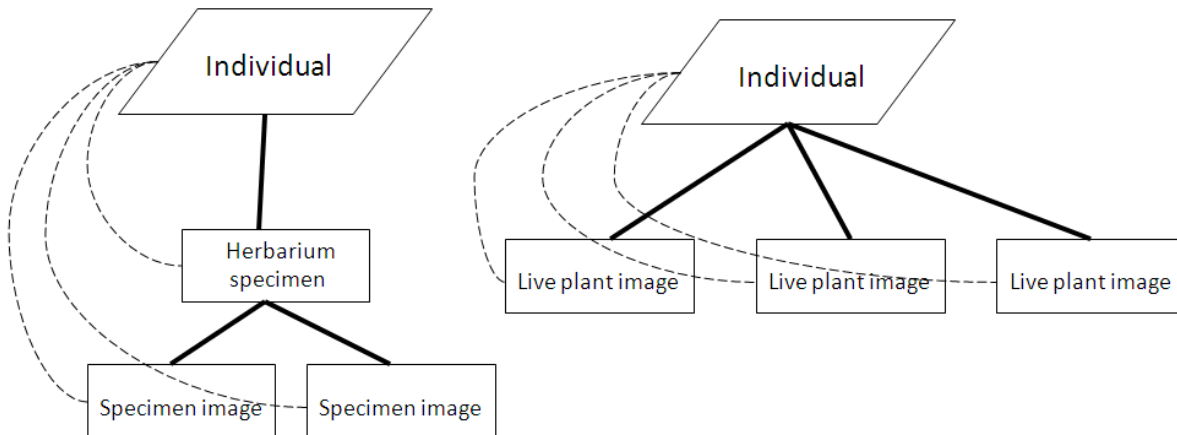
1. Conflict of interest between live plant photographers and specimen databasers.

In the paper [Organization of occurrence-related biodiversity resources based on the process of their creation and the role of individual organisms as resource relationship nodes](#). (Baskauf, S.J.

2010. *Biodiversity Informatics* in press) I make the case for an organizational system for biodiversity resources that is based on the way the resources are created. That system is focused on the role of the Individual source organisms as a means to connect multiple Occurrences. The advantage of this system is that it allows a single system for organizing biodiversity resource metadata that can handle even the most complex relationships among resources and can deal with multiple determinations (see the figure below).



A system that recognizes the role of the Individual source organism is crucial for live plant images, where there are nearly always several Occurrences (images) per individual (below right). On the other hand, it is not very important for herbarium specimens and their images (below left) because there is often only a single specimen associated with an individual. In that case, creating and databasing a separate guide for the individual is superfluous. (However, it should also be apparent from these diagrams that linking specimens to a resource representing the source Individual would be an appropriate way to link duplicate specimens.)



So the challenge in creating a system for identifiers and a structure for metadata is to do it in such a way that it allows for the more complicated system required by live plant images, but does not introduce unnecessary complexity and extra data management for specimen curators that do not require it.

Fortunately, there is a solution to this problem that allows specimen databasers to ignore Individuals if they wish while still making their metadata structure compatible with metadata of live plant images. I will start with some conventions and then explain the strategy, followed by examples.

2. Default guids for source Individuals and images of specimens when not explicitly assigned.

Beyond the general recommendations for the construction of HTTP URI guids in section II.C., I recommend the following conventions:

- For an Occurrence identified by a non-hash URI (i.e. a "[303 URI](#)") which does not have source Individual explicitly identified with its own HTTP URI, by convention the URI of the source Individual will be the Occurrence URI plus the fragment identifier "#ind".
- For an Occurrence identified with a non-hash URI which has a single image representation that is not explicitly identified by its own HTTP URI, by convention the URI of the image will be the Occurrence URI plus the fragment identifier "#img".
- Occurrences identified with hash URIs may not use the fragment identifiers "#img" and "#ind".
- For an Occurrence identified using a hash URI, and which does not have a source Individual explicitly identified by its own HTTP URI, by convention the URI of the source individual will be the base Occurrence URI without its fragment identifier plus the fragment identifier "#ind".
- For an Occurrence identified with a hash URI, and which has a single image representation that is not explicitly identified by its own HTTP URI, by convention the URI of the image will be the base Occurrence URI without its fragment identifier plus the fragment identifier "#img".

The following examples illustrate these rules.

Example 1. A herbarium assigning locally unique bar codes of the form "hb123456" to specimens creates HTTP URI guids of the form

```
http://herbarium.org/hb123456
```

for the physical specimen itself. By the suggested convention, the source plant would have the URI

```
http://herbarium.org/hb123456#ind
```

The specimen image would have the URI

`http://herbarium.org/hb123456#img`

In the institution's local database, it is not necessary to use any identifier beyond the bar code string since all three of the guids above can be constructed from the bar code by simple rules.

Example 2. A herbarium using accession numbers creates a locally unique identifier from the collection year and an accession number that starts with zero on Jan 1. To clarify that their HTTP URI represents a non-information resource, they decided to append the fragment identifier "#specimen" to their URIs. (There is nothing in the rules for HTTP URI guids that compels them to do this - it is a matter of preference.) So the 3422rd specimen collected in 2010 was assigned the guid

`http://otherherbarium.edu/2010/3422#specimen`

By the suggested convention, the source plant would have the URI

`http://otherherbarium.edu/2010/3422#ind`

The specimen image would have the URI

`http://otherherbarium.edu/2010/3422#img`

3. Rule for constructing HTTP URIs for determinations of an individual or specimen when one is not explicitly assigned

Here I use the term "determination" to indicate an abstract resource describing the assignment of a taxonomic identity to an individual or the specimen. This term carries the same meaning as the Darwin Core class Identification (<http://rs.tdwg.org/dwc/terms/index.htm#Identification>), but I choose to use the term determination because such a resource might include any of the following: initial identifications, annotations, or assignments of multiple identities based on different taxon concepts.

If a record had only a single determination, it would not be necessary to assign a guid to the determination because the determination metadata could simply be included in the record for the individual or specimen. However, since there may be an initial identification followed by one or more annotations, or since a particular organism might be assigned to different taxa depending on the concept followed, there needs to be some way (e.g. a URI) to differentiate among the metadata terms associated with a particular determination. Assignment of an HTTP URI guid to each determination is also important because the [Linked Data recommendations](#) discourage the use of blank nodes.

Since each determination can only be associated with one individual, it makes sense to base the URI of the determination on the URI of the individual using a hash URI. This can be done by creating a fragment identifier from a unique identifier for the taxon or taxon concept assigned by the determination (as long as the taxon identifier system is kept consistent for a particular individual). For example, if the specimen `http://herbarium.org/hb123456` was determined to be *Quercus alba*, which has an ITIS taxonomic serial number of 19290, the

determination that assigned the taxonomic identity of *Quercus alba* to the specimen would have the URI

<http://herbarium.org/hb123456#19290>

If the herbarium used taxon concepts specified by geospecies.org in which *Quercus alba* has the unique identifier **waK4b**, the specimen would have the URI

<http://herbarium.org/hb123456#wak4b>

It is also possible to assign completely independent URIs (i.e. not based on the URI of the specimen) to the determinations, but using a rule to generate the URIs reduces the amount of record-keeping required.

4. Terms used to indicate the relationship among resources.

The need for metadata terms to express how one resource or one database record is related to another will depend on the complexity of the database. In a simple database where every record can be expressed as one row in a table, there are few terms needed to express how one record is related to another. An example of such a simple database based on Darwin Core is described at <http://rs.tdwg.org/dwc/terms/simple/index.htm>. However, in a more complex database, ID reference fields (idrefs) may be needed in a record to associate a particular record in one table to a related record in another table. In RDF, some metadata terms (acting as "predicates") serve the role of indicating the nature of the connection between the resource that is the subject of the RDF and other resources.

Darwin Core contains a number of terms that can be used as described in the previous paragraph. Some of those terms are:

dwc:individualID

dwc:identificationID

dwc:taxonConceptID (or *dwc:taxonID*)

dwc:associatedMedia

(See <http://rs.tdwg.org/dwc/terms/index.htm> for term definitions.) There are also some terms that have meaning in a general context beyond the biodiversity informatics community:

foaf:depicts

foaf:depiction

bibo:Webpage

foaf:isPrimaryTopicOf

owl:sameAs

Some terms in this latter category express similar relationships to those in the Darwin Core list. However, it may still be advisable to include them in an RDF description of a resource because that would allow linked data clients outside the biodiversity community to understand the relationship.

The individual-based organizational system that I have suggested (Baskauf 2010) makes use of the Darwin Core relational terms, but adds some other terms that are necessary to express relationships that are missing from Darwin Core:

sernec:derivedFrom

sernec:derivativeOccurrence

sernec:identifiedIndividual
sernec:basedOnOccurrence
serenc:usedInDetermination

The definitions of these terms are at <http://bioimages.vanderbilt.edu/rdf/terms> . Along with the other relational terms above, they will be used in subsequent examples.

5. Conceptual representation of images.

Unlike specimens, which are physical resources that cannot be delivered via the Internet, digital images are clearly information resources that can be delivered. As an information resource, the digital image itself could be considered "data" (i.e. an immutable series of bytes) to be delivered when there is a request for the resolution of its URI (in other words, the URI could also be acting as a URL). In addition to the image itself, the metadata about the image could also be examined by a user. Although there is nothing conceptually wrong with this approach, there are several practical problems with it.

The first problem is that this approach causes image GUIDs to behave differently than GUIDs for other biodiversity resources. When users type the GUID of a specimen into a web browser, they expect to get a web page providing metadata about the specimen. However, when users type a GUID which is the URL of an image of a live plant into a web browser, they get the digital image, not a web page providing metadata. How do they ever get that metadata? The second problem is that an image may have several forms (a high-resolution version, a web-resolution version, and a thumbnail) that all share the same *dcterms:Location*, *dwc:recordedBy*, and *dwc:eventDate* metadata. It would not make sense to repeat those metadata several times. The third problem is that if the URL of the image is intended to be a GUID, it can't change. That means that if the GUID is the URL of the image, the image provider would be stuck with keeping the image at the same URL forever and wouldn't have the option of moving it to a different repository under some other domain name.

The solution to all of these problems is to consider the image to be an abstract resource (i.e. an abstract thing representing what is captured when the photographer presses the shutter on a camera) that can have several representations. If the generic image and each of these representations are considered to be abstract, they can be assigned unchanging HTTP URIs that are independent of the URLs that are actually used to retrieve the digital images themselves.

The [Media Resources Task Group \(MRTG\) schema](#) defines a class called [Service Access Point](#) which describes network access to a media resource described by metadata. Service Access Points have the property *mrtg:variant* which can describe the quality of the version of the media that is accessed by that Service Access Point. Some accepted values are: "Thumbnail", "Lower Quality", "Medium Quality", "Good Quality", and "Best Quality".

I recommend the following convention for GUIDs for images and their service access points.

The generic image as an abstract resource is identified by the base URI, e.g.

<http://bioimages.vanderbilt.edu/baskauf/66921>

Each service access point should have an HTTP URI guid formed by concatenating the base URI and a fragment identifier, e.g.

<http://bioimages.vanderbilt.edu/baskauf/66921#tn> for the thumbnail access point and

<http://bioimages.vanderbilt.edu/baskauf/66921#bq> for the original high resolution access point. These GUIDs are not the URLs of the images. The actual URL from which the image can be retrieved is the value of the term *mrtg:accessURL* that is a property of the service access point identified by the HTTP URI.

Three points can be made about the service access points. One is that GUIDs are required to be "semantically opaque". That means that one must not depend on an interpretation made from the structure of the identifier. For example, one should not assume that a URI with a "tn" fragment identifier is a thumbnail. Rather, one should retrieve the value of *mrtg:variant* for that service access point and determine whether or not its value is "Thumbnail". The second point is that if the digital image is data, the rules of GUIDs demand that the data not change once the identifier has been assigned. This is required so that a user can be confident that data retrieved with a particular identifier is the same ten years from now as it is in the present. If the file containing the image is changed, then technically a new version of the identifier should be created. Given the impracticality of this (i.e. the burden of tracking and maintaining a multitude of image versions), it is better not to assign GUIDs to images until they are processed and a clear decision has been made as to what files will be made available for particular variants. The third point is that there is no requirement that the various Service Access Points be stored as separate image files. The lower quality versions could be generated dynamically by software from the original Best Quality image. In this case, the *mrtg:accessURL* might contain a query string telling software on the server how to construct the lower quality image. Separating the accessURL from the HTTP URI for the Service Access Point allows one to change from one method of providing image versions (static files vs. dynamic generation) without violating the rule requiring that GUIDs do not change.

The choice of which fragment identifiers to use and how many variants should be made available is up to the data provider. However, the SERNEC Live Plant Imaging Group recommends that the high resolution versions be made available for all images in its Basic collection and requires that high resolution versions be made available for all images in its Full collection. In addition, for the purposes of creating indices, I recommend also always making available a thumbnail 100 pixels in its longest dimension.

Suggested versions for image Service Access Points are:

example URI variant	value of <i>mrtg:variant</i>	pixels in longest dimension	Comment
http://domain.org/12345#tn	Thumbnail	100	
http://domain.org/12345#lq	Lower Quality	480	iPhone screen size
http://domain.org/12345#gq	Good Quality	1024	standard screen size
http://domain.org/12345#bq	Best Quality	3000 or more	original image

See http://www.keytonature.eu/wiki/Submission_v0.9#Variant for recommendations about variants. The details for structuring the metadata of Service Access Points will be discussed below.

6. Strategy for simultaneously accommodating simple and complex models for occurrence metadata relationships

The operative principles underlying this strategy are:

- Metadata providers that do not require a complex metadata framework (e.g. herbarium curators) will not be forced to adopt one. Rather they can use a simple "flat" database structure if they want. They also will need to create only one guid to identify a single specimen.
- Metadata providers that desire or need a complex metadata framework (e.g. live plant photographers) will be able to utilize the more complex system I described in my paper (Baskauf 2010). They can create multiple independent guides to identify the various resources in their database.
- Both approaches to metadata structure will be usable under the system described in Baskauf 2010, while specimen metadata will be understandable by linked data clients that don't care about live plant images and that don't know about , understand, or accept the Baskauf 2010 system.

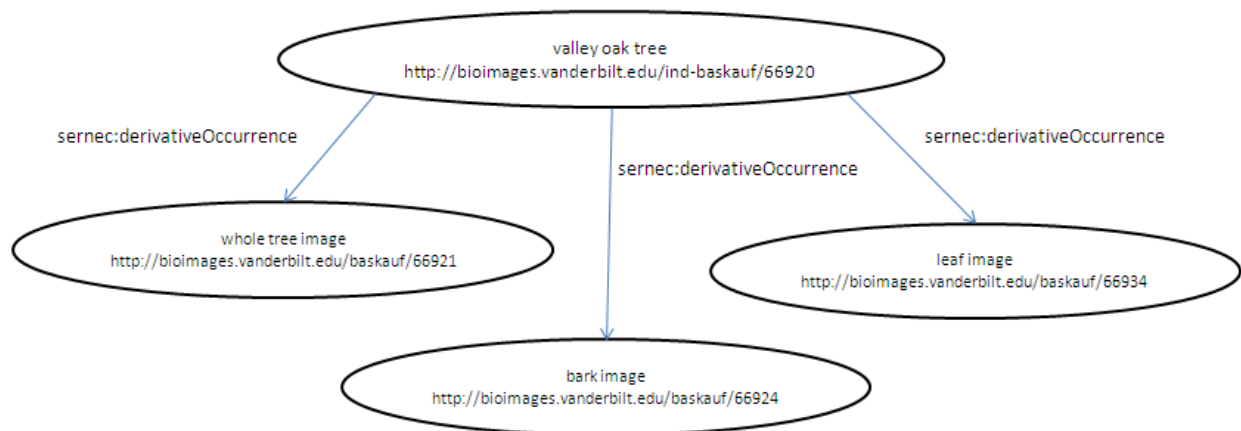
The way these principles will be achieved is through the HTTP URI construction rules described previously and through the method of structuring of the RDF files associated with the URIs as described below.

7. General structure of RDF files when Individuals are assigned URIs independently from their associated Occurrences (images and specimens)

The RDF describing the Individual and each Occurrence associated with the Individual will be located in separate XML files, with one file for each HTTP URI guid. The following example illustrates this structure using actual functional URIs and RDF from Bioimages. The Individual is the *Quercus lobata* (valley oak) tree having the guid

<http://bioimages.vanderbilt.edu/ind-baskauf/66920>

The graph below shows the Individual and three Occurrences derived from it (an image of the whole tree <http://bioimages.vanderbilt.edu/baskauf/66921>, an image of the bark, and an image of a leaf).



The overall structure of the RDF file for the Individual is:

Identifier for the Individual (<http://bioimages.vanderbilt.edu/ind-baskauf/66920>)

Metadata about the Individual

Terms linking the Individual to other resources

Identifier for the Determination (<http://bioimages.vanderbilt.edu/ind-baskauf/66920#19370>)

Metadata for the Determination

Terms linking the Determination to other resources

[more Determinations if necessary]

(dashed lines separate elements in the RDF file that are identified by a different GUID). The overall structure of the RDF file for the Occurrence is:

Identifier for the Occurrence (<http://bioimages.vanderbilt.edu/baskauf/66921>)

Metadata about the Occurrence

Terms linking the Occurrence to other resources

Identifier for a Service Access Point if an image (<http://bioimages.vanderbilt.edu/baskauf/66921#tn>)

Metadata about the Service Access Point

Term (*accessURL*) linking the Service Access Point to the file URL

[more Service Access Points if necessary]

In XML format, the basic structure of the RDF file for the individual is:

```
<rdf:Description rdf:about="http://bioimages.vanderbilt.edu/ind-baskauf/66920">
  <rdfs:type rdf:resource="http://bioimages.vanderbilt.edu/rdf/terms#Individual"/>
  ... information about the tree ...
  <dwc:identificationID rdf:resource="http://bioimages.vanderbilt.edu/ind-baskauf/66920#19370"/>
  <sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66921"/>
</rdf:Description>
<rdf:Description rdf:about="http://bioimages.vanderbilt.edu/ind-baskauf/66920#19370">
  <rdfs:type rdf:resource="http://rs.tdwg.org/dwc/terms/Identification" />
  ... information about the determination ...
  <sernec:identifiesIndividual rdf:resource="http://bioimages.vanderbilt.edu/ind-baskauf/66920"/>
</rdf:Description>
```

(See Appendix C for the entire file.)

The *rdf:about* attributes identify the resources described in the two parts of the file (the individual tree and the determination). The *rdfs:type* term describes the kind of thing that the resource is. The *dwc:identificationID* term specifies that the tree has the determination of TSNID 19370. The *sernec:derivativeOccurrence* term specifies that the tree has the Occurrence which is the whole tree image. The *sernec:identifiesIndividual* term specifies that the determination identifies the tree. (Note: there is some lack of clarity about exactly how biodiversity resources should be typed using *rdfs:type*. I have chosen to type resources by their

Darwin Core classes, if one exists for a particular resource. However, ultimately the biodiversity community may reach some other consensus. The type of the service access point [below] is defined implicitly by use of *mrtg:hasServiceAccessPoint* as the container element.)

Here is the basic structure (abbreviated) of the RDF file for the occurrence:

```
<rdf:Description rdf:about="http://bioimages.vanderbilt.edu/baskauf/66921">
  <rdfs:type rdf:resource="http://rs.tdwg.org/dwc/terms/Occurrence"/>
  ... information about the image ...
  <dwc:individualID rdf:resource="http://bioimages.vanderbilt.edu/ind-baskauf/66920"/>
  <sernec:derivedFrom rdf:resource="http://bioimages.vanderbilt.edu/ind-baskauf/66920"/>
  <mrtg:hasServiceAccessPoint rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66921#tn"/>
  <mrtg:hasServiceAccessPoint rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66921#bq"/>
</rdf:Description>
<mrtg:hasServiceAccessPoint rdf:about="http://bioimages.vanderbilt.edu/baskauf/66921#tn">
  ... information about the thumbnail service access point ...
  <mrtg:accessURL>http://bioimages.vanderbilt.edu/tn/baskauf/t66921.jpg</mrtg:accessURL>
</mrtg:hasServiceAccessPoint>
<mrtg:hasServiceAccessPoint rdf:about="http://bioimages.vanderbilt.edu/baskauf/66921#bq">
  ... information about the best quality service access point ...
  <mrtg:accessURL>http://bioimages.vanderbilt.edu/baskauf/66921.jpg</mrtg:accessURL>
</mrtg:hasServiceAccessPoint>
```

(See Appendix D for the entire file.)

8. General structure of RDF files for specimens that have a single image

There are two primary differences between the format of the RDF data for metadata of individuals having GUIDs that are independent of the GUIDs of their Occurrences and the format of RDF metadata for specimens having a single image. One is that the metadata are located in a single file. The other is that it will be assumed that (at least initially) the Individual has a single derived Occurrence (the specimen) which in turn has a single Occurrence derived from it (the specimen image).

These differences are consistent with the HTTP URI construction conventions described in section E.2. The use of fragment identifiers ("#ind" and "#img") to differentiate the Individual and the specimen image from the specimen itself requires that the metadata be in a single file identified by the root of the URI. The naming conventions also require that there be only a one-to-one relationship (rather than one-to-many) between the Individual, specimen, and specimen image described in that file.

The graph below shows the specimen, the image derived from it, the Individual from which the specimen was derived, and that Individual's determination (assuming that the Individual was a white oak tree having ITIS TSN 19290):



The overall structure of the RDF file would look like this

Identifier for the Individual (<http://herbarium.org/hb123456#ind>)

Metadata about the Individual

Terms linking the Individual to the specimen and determination

Identifier for the Determination (<http://herbarium.org/hb123456#19290>)

Metadata for the Determination

Term linking the Determination to the individual

Identifier for the specimen Occurrence (<http://herbarium.org/hb123456>)

Metadata about the specimen Occurrence

Terms linking the specimen Occurrence to other resources

Identifier for the specimen image Occurrence (<http://herbarium.org/hb123456#img>)

Metadata about the Occurrence

Terms linking the specimen image Occurrence to other resources

Identifier for the best quality image Service Access Point (<http://herbarium.org/hb123456#bq>)

Metadata about the Service Access Point

Term (*accessURL*) linking the Service Access Point to the file URL

In XML format, here is the basic structure (abbreviated) of the combined RDF file:

```
<rdf:Description rdf:about="http://herbarium.org/hb123456#ind">
  <rdfs:type rdf:resource="http://bioimages.vanderbilt.edu/rdf/terms#Individual"/>
  ... information about the individual ...
  <sernec:derivativeOccurrence rdf:resource="http://herbarium.org/hb123456"/>
  <dwc:identificationID rdf:resource="http://herbarium.org/hb123456#19290"/>
</rdf:Description>

<rdf:Description rdf:about="http://herbarium.org/hb123456#19290" >
  <rdfs:type rdf:resource="http://rs.tdwg.org/dwc/terms/Identification"/>
  ... information about the determination ...
  <sernec:identifiesIndividual rdf:resource="http://herbarium.org/hb123456#ind"/>
</rdf:Description>

<rdf:Description rdf:about="http://herbarium.org/hb123456">
  ... information about the specimen itself ...
  <dwc:individualID rdf:resource="http://herbarium.org/hb123456#ind"/>
  <dwc:identificationID rdf:resource="http://herbarium.org/hb123456#19290"/>
  <sernec:derivedFrom rdf:resource="http://herbarium.org/hb123456#ind"/>
  <sernec:derivativeOccurrence rdf:resource="http://herbarium.org/hb123456#img"/>
  <dwc:associatedMedia rdf:resource="http://herbarium.org/hb123456#img"/>
</rdf:Description>

<rdf:Description rdf:about="http://herbarium.org/hb123456#img">
  <rdfs:type rdf:resource="http://rs.tdwg.org/dwc/terms/Occurrence"/>
  ... information about the specimen image ...
  <dwc:individualID rdf:resource="http://herbarium.org/hb123456#ind"/>
  <sernec:derivedFrom rdf:resource="http://herbarium.org/hb123456"/>
  <mrtg:hasServiceAccessPoint rdf:resource="http://herbarium.org/hb123456#bq"/>
</rdf:Description>

<mrtg:hasServiceAccessPoint rdf:about="http://herbarium.org/hb123456#bq">
  ... information about the best quality service access point ...
  <mrtg:accessURL>http://herbarium.org/images/dsc55794.jpg</mrtg:accessURL>
</mrtg:hasServiceAccessPoint>
```

(See Appendix B for the entire file.)

The following points about the RDF should be noted:

1. Although there are five different HTTP URI guids in this RDF metadata, they are all created from the basic URI of the specimen by fixed rules that specify hash strings to be concatenated to the end of the base URI. The "rule" of Linked Data and the Semantic web that different "things" must be represented by different URIs has been followed, but without requiring the herbarium to keep up with five different identifiers. All of the derived URIs are ultimately based on a local identifier, the barcode of the specimen.
2. The structuring of these metadata require a linked data client (i.e. a computer program trying to figure out what is going on by looking at the RDF) to understand the Darwin Core and MRTG schemas, but does not require it to understand terms in the sernec: namespace or to accept the resource structuring system described in Baskauf 2010. Thus metadata constructed this way would be compatible with both a system designed around the conceptual framework of Baskauf

2010 and a "flatter" system that does not accept the concept of Individuals as organizational nodes.

3. In particular, applying the property

```
<dwc:identificationID rdf:resource="http://herbarium.org/hb123456#19290"/>
```

to both the specimen and individual allows one to take either the outlook that determinations should be associated with specimens or that determinations should be associated with the individuals from which the specimens are derived.

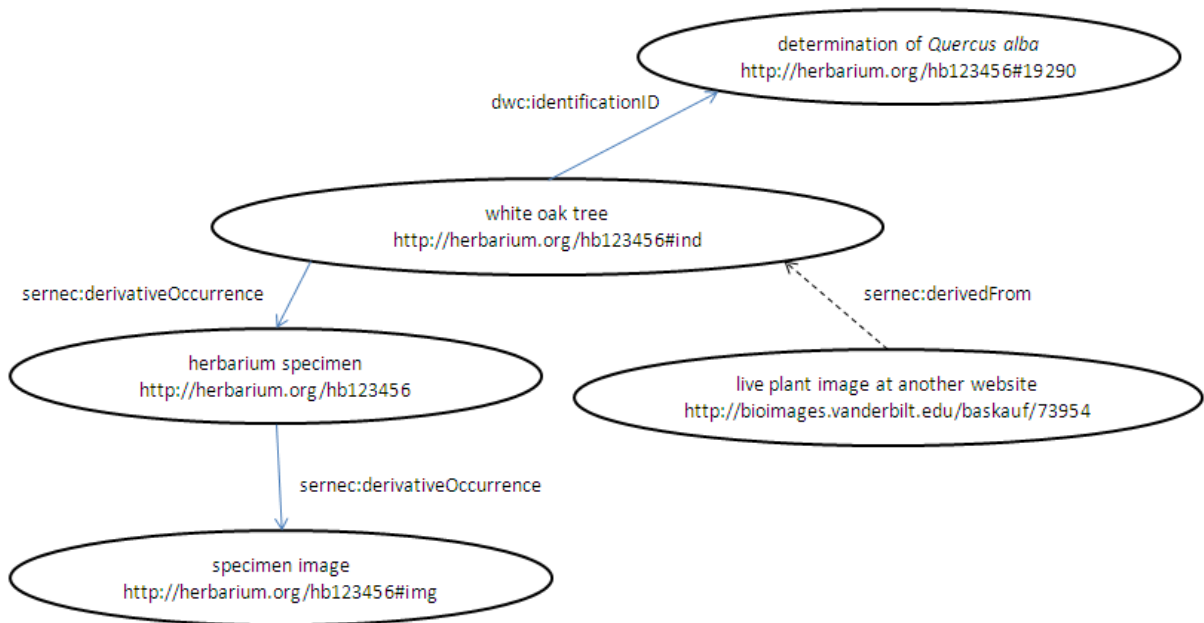
4. This structure could also accommodate an even more "stupid" linked data client that could only perceive the "flattest" metadata structure if terms from the Darwin Core Taxon class were used to specify properties of the specimen, e.g.

```
<rdf:Description rdf:about="http://herbarium.org/hb123456">
  <rdfs:type rdf:resource="http://rs.tdwg.org/dwc/terms/Occurrence"/>
  <dwc:family>Fagaceae</dwc:family>
  <dwc:genus>Quercus</dwc:genus>
  <dwc:specificEpithet>alba</dwc:specificEpithet>
  ... etc. ...
  ...more information about the specimen itself ...
  <dwc:individualID rdf:resource="http://herbarium.org/hb123456#ind "/>
  <dwc:identificationID rdf:resource="http://herbarium.org/hb123456#19290"/>
  <sernec:derivedFrom rdf:resource="http://herbarium.org/hb123456#ind"/>
  <sernec:derivativeOccurrence rdf:resource="http://herbarium.org/hb123456#img"/>
  <dwc:associatedMedia rdf:resource="http://herbarium.org/hb123456#img"/>
</rdf:Description>
```

The more elegant system of linking (potentially several) determinations to the Individual via the *dwc:identificationID* term is more flexible, but supplying the Taxon class terms within the specimen element is a bet-hedging strategy that increases the probability that a linked data client will actually understand the taxonomic identity of specimen resource.

9. Advantage of this approach #1: Ability of "foreign" authorities to link to guides

One of the basic principles of the Linked Data/HTTP URI guid concept is that guides issued by one data provider are also used by other data providers (rather than having the other data providers create new guides for the same resource). For example, if at the time the white oak specimen were collected (or at any subsequent time for that matter) I were to take images of the white oak tree from which the specimen were taken, I could link my images to the specimen like this (assuming that I knew the barcode of the specimen and the system used by the herbarium to construct its guides):



The actual RDF XML used to do the linking would be:

```
<rdf:Description rdf:about="http://bioimages.vanderbilt.edu/baskauf/73954">
  <rdfs:type rdf:resource="http://rs.tdwg.org/dwc/terms/Occurrence"/>
  ... information about the oak tree image ...
  <sernec:derivedFrom rdf:resource="http://herbarium.org/hb123456#ind"/>
  <dwc:individualID rdf:resource="http://herbarium.org/hb123456#ind"/>
  <foaf:depicts rdf:resource="http://herbarium.org/hb123456#ind"/>
</rdf:Description>
```

There are several points related to this example:

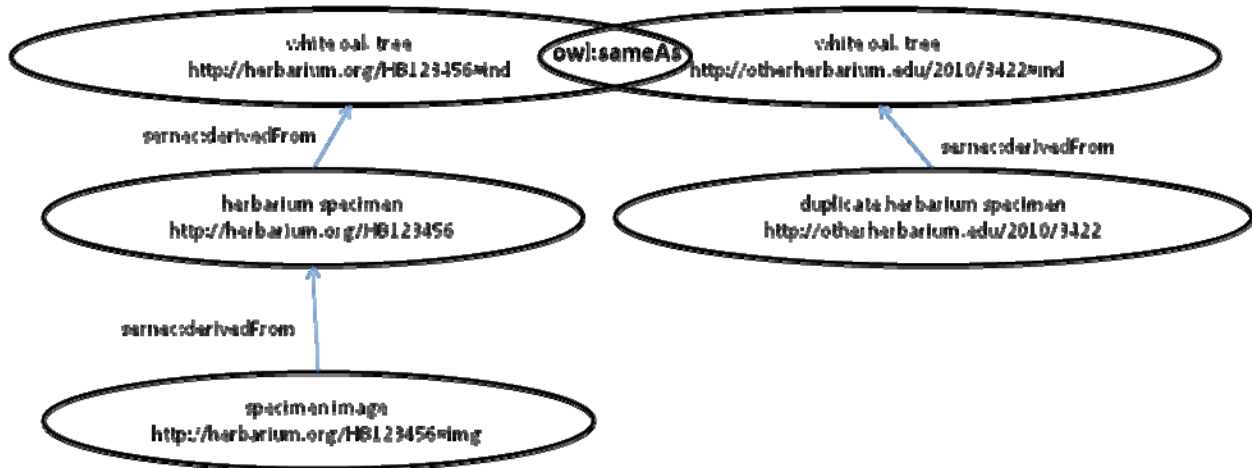
1. The three terms used to link the live plant image to the oak tree assume varying degrees of understanding on the part of the linked data client that would be examining the RDF:
 - The least savvy type of client would probably know what *foaf:depicts* means since the FOAF vocabulary is widely used. About all they would learn from dereferencing its object (the URI for the oak tree individual) would be that the image is taken of some physical object (if the term *dcterms:type* value of the individual were set to *PhysicalObject*) having the description given to the individual using the *dcterms:description*.
 - A more savvy client that knew about Darwin Core would know all of the things that the least savvy client knew, plus that the image was of an individual organism (based on the use of the *dwc:individualID* term). The more savvy client could also resolve the *dwc:identificationID* property of the individual to find out what kind of organism it was.
 - The most savvy client that knew about Darwin Core and the Baskauf 2010 organizational system would know everything that the more savvy client knew, plus would know that there was also a specimen at herbarium.org whose URI could be dereferenced to find out even more about the oak tree. Assuming that an image of the specimen were available, the most savvy client could even present that specimen image on the screen with the live images.
2. Although it would be desirable for the specimen metadata provider to link the individual from which the specimen was derived to the live plant image by using the

sernec:derivativeOccurrence and *foaf:depiction* terms in the specimen RDF XML file, that is not required in order for a linked data client to understand how the live plant image is related to the individual. As long as the linked data client has "discovered" the image, it can use knowledge that the *foaf:depicts* and *sernec:derivedFrom* terms are inverse properties of *foaf:depiction* and *sernec:derivativeOccurrence* properties respectively to infer the relationship of the individual tree to the live plant image. Thus there is no burden introduced on specimen data providers in making it possible for their guides to be used by others, other than the minimal effort exerted to structure their RDF metadata appropriately. [Note: currently the properties *sernec:derivedFrom* and *sernec:derivativeOccurrence* are not defined in Web Ontology Language (OWL), so they don't yet have the property *owl:inverseOf* which would be necessary for a linked data client to "understand" that they are inverse properties. The FOAF properties are related by the *owl:inverseOf* property, however.]

3. By not explicitly providing a determination for the live plant image, the live plant photographer can defer the identification of the image to the taxonomist responsible for the specimen metadata. Identification information about the image can be provided to "stupid" linked data clients through the *dcterms:title* property of the image, but "smarter" clients could make the connection (through the Individual metadata) to one or more determinations and then create a human-readable web page that provides information on subsequent annotations of an original identification, or on multiple opinions of identity based on different species concepts.

10. Advantage of this approach #2: Linking duplicate specimens

If it were discovered that a duplicate specimen from the oak tree were part of the collection at a different herbarium, that herbarium could link to this specimen by declaring that the two individual organism guides (i.e. the tree guides) identify the same thing:



This would be accomplished by the following RDF:

```
<rdf:Description rdf:about="http://otherherbarium.edu/2010/3422#ind">
  <rdfs:type rdf:resource="http://bioimages.vanderbilt.edu/rdf/terms#Individual"/>
  ... information about the individual ...
  <owl:sameAs rdf:resource="http://herbarium.org/hb123456#ind"/>
  <sernec:derivativeOccurrence rdf:resource="http://otherherbarium.edu/2010/3422"/>
  <dwc:identificationID rdf:resource="http://otherherbarium.edu/2010/3422#19290"/>
</rdf:Description>
etc.
```

If a linked data client "discovered" the records in both herbaria, it would automatically make the connection that the two specimens were duplicates by virtue of the fact that one record declared the source individuals to be the same. Any information that was known about the individual oak trees would be merged. For example, if the specimens in the two herbaria were annotated by different taxonomists and each taxonomist determined that the tree was a member of a different taxon, then both determination opinions would be associated with the tree in the combined database created by the linked data client, even though the determinations were recorded in different RDF files at different institutions. In addition, each specimen would be listed as a *sernec:derivativeOccurrence* of the oak tree.

F. Overcoming Barrier 4: Figuring out how to implement the delivery of the HTML and RDF files

1. RESTful services.

The World Wide Web is built around a concept called "Representational State Transfer" (REST). If you are interested in knowing about the details of REST, you can read the [Wikipedia article](#). Here I will briefly explain how the concept of REST applies to the resolution of GUIDs.

The consumer of the metadata about a resource is called the "client". The client is a computer program that wants information about the resource and could be a web browser, RDF browser, Google indexing robot, or some kind of specialized computer software that is building a database of images and specimens. The provider of the metadata about a resource is called the "server". The server is a networked computer that is capable of responding to requests for information through the Internet. The client uses specific language (the HTTP protocol) to tell the server what information it wants. The most common HTTP request is GET, which simply tells the server that the client would like to get a particular resource. The client describes the exact resource that it wants by means of the HTTP URI of the resource (in this case the GUID of a specimen, image, individual, determination, etc.).

An important thing here is that neither the client nor the server needs to know the purpose of the exchange. The client does not know or care how the server produces the file that it sends. The server does not know or care what the client plans to do with the information that the server sends. This separation of concerns means that the programmers designing client software don't have to worry about how the server software is designed or the exact mechanism used by the

server to produce the files. One day the server might use one kind of software and method to generate the files it serves and the next day that method might change, but the client would never know the difference. The designers and managers of the server software do not need to worry about how the clients are going to use the data they send. They simply send the files and the usefulness of those files depends on the cleverness of the design of the client software.

2. Representations.

A URI (uniform resource identifier) that is a URL (uniform record locator) provides the address from which a particular file will be served. For example:

<http://bioimages.vanderbilt.edu/ind-baskauf/66920.rdf>

is the URL for the RDF formatted XML file that provides the metadata describing a valley oak tree in Mt. Diablo State Park in California. In contrast, the URI

<http://bioimages.vanderbilt.edu/ind-baskauf/66920>

identifies the valley oak tree itself. The oak tree cannot be sent from a server to a client in the same way that the RDF file can. If a client sends an HTTP GET request for the valley oak URI, the server will have to figure out a representation of the oak tree to send to the client, since it can't send the tree itself. In the context of guides, the two common representations would be a web page describing the tree (in HTML format) and RDF metadata describing the tree (in XML format).

In an interaction of this sort, the client can tell the server what kind of representation it prefers by using a "MIME" type. The MIME type for a web page is "text/html" and the MIME type for RDF in XML format is "application/rdf+xml". The preferred representation (in the form of a MIME type) of the resource is sent to the server as a part of the GET request. After the client requests a particular representation for a physical or abstract resource, there are three possible exchanges with the server that can transpire:

1. The server replies with a "404 Not Found" HTTP response because it doesn't have a file with that URI. This represents failure on the part of the metadata provider and its server administrator and is unacceptable.
2. The server replies with a "303 See Other" HTTP response which gives the (different) URL of the document of the type requested by the client. The client then sends the server a request for that document and the server sends the representation requested by the client. This method of interaction is called "[content negotiation](#)" and is probably the best method for resolving the URI.
3. The server replies with a "200 OK" HTTP response and sends a document to the client without paying attention to the MIME type requested by the client. This document may or may not be the kind that the client wants. This method of interaction may be OK if the document itself contains a link to the type of representation that the client wants AND if the client knows how to follow the link to the needed representation. I will call this method of interaction the "**link**" **method**. (An example of this method is given in Appendix 1 of <http://www2.gbif.org/Persistent-Identifiers.pdf>.)

We will make the assumption that any useful client software is able to use either the content negotiation method or the link method to get the kind of representation it wants (HTML or

RDF). In that case, **as long as the HTTP URIs used by the data provider have an appropriate format**, the provider does not need to commit permanently to any particular method of providing access to the derived representation. The data provider could utilize whatever method is the easiest to implement given the IT resources that it has available at the present and could change to the other method at some point in the future if the circumstances change. This design is RESTful in the sense that the client doesn't need to concern itself with the resource constraints of the server and that the server can assume that the client will be able to get what it needs by either method, i.e. the problems of the client and the server are their own business; they do not have to concern themselves with the problems of the other party.

3. Methods of file serving.

There are two basic methods that can be used by the server to serve the necessary files to the client.

Static file method. One method is to create two **static** files (one RDF and one HTML) for every guid that the server is responsible for resolving. The advantage of this method is that it is very simple. It can be implemented on a generic web server and does not require any special server-side software to be running. It also does not require any IT expertise or long-term maintenance in order to provide the resolution services required by the rules governing HTTP URI guides. The disadvantage of this method is that a large database requires a large number of files and any changes made to the types of metadata provided require regenerating all of the files. Also, changes made to the data provider's database will only show up in the guid resolution as often as the provider updates the static files.

Dynamic method. The other method is to run specialized web server software that uses the data provider's database to generate the files needed for guid resolution at the time the files are requested by the client. The advantage of this method is that any changes made to the provider's database could be immediately reflected in the metadata provided during guid resolution. There would be few additional files required on the server beyond the files already present in the database. The disadvantage of this method is that the specialized software would have to be programmed or at least set up by an IT professional with specialized knowledge of such systems. Resolution of the guides would be dependent on continued maintenance of both the database and the specialized web server software. If either of those components went offline for an extended period of time, the guides would fail the "persistence" requirement.

The relative desirability of these two methods depends on the number of guides being maintained, the frequency at which changes to the metadata occur, and the ability of financial and IT resources to the data provider. The static file method would probably be the best for collections having relatively few records (perhaps tens of thousands or fewer), collections that rarely changed, or for institutions with little or no IT support. The dynamic method would probably be the best for collections having many records (perhaps hundreds of thousands or more), institutions with frequent changes and metadata updates, and for institutions with good IT support available.

4. Long-term flexibility for the data provider.

Because of the RESTful nature of HTTP URIs, it is not necessary for the client to know or care which of the two methods are being used by the server. The client simply requests URIs and receives files and the means by which this is accomplished is not important to the client. This is advantageous to the data provider, because it means that a data provider can change from one method to another **as long as the HTTP URIs used by the data provider have an appropriate format.**

Normally one would assume that the progression would be from the static method to the dynamic method because one would assume that the conditions of the data provider would progress from small to larger collections, that IT resources and support would increase and have costs that decrease over time. However, one should also consider the possibility that over the long term, collection data activity might go down, and IT resources and support could go down (i.e. through loss of funding or grant support, through merger or elimination of departments, through loss of faculty or staff positions, through loss of support from administration, etc.). In this case, a data provider that was once serving metadata files dynamically might be forced to resort to serving static files on a generic web server. This could probably be feasible even for a very large number of static files (perhaps a million or more) if the files rarely or never changed and if the directory structure imposed by the URI format could handle that many files.

5. Relationship between HTTP URI format and maintaining flexibility of file generation method and method of providing representation.

The recommendations for construction of HTTP URI guides outlined in section II.C. were written considering the need for maximum flexibility in server methodology. This consideration is particularly important because of the requirement for long-term stability of guides. Since the circumstances of a data-providing institution are likely to change over the long term, it would be unadvisable to structure the guides in such a way that would lock the institution into a particular method of providing the metadata which must be served when the guides are resolved.

In order to allow flexibility in the method used to provide representations of abstract URIs, it is advisable for the URIs to not have a file extension. This makes the URI independent of any particular file type and makes it possible at some point in the future to provide resource representations that have not yet been conceived.

In order to allow flexibility in the method of file generation, it is advisable to use only characters that are unlikely to cause errors in any known system of storage or delivery (see section II.C.3.).

In order to limit the number of static files that might need to be placed in a single directory, using the *namespace + "/" + objectIdentifier* system was recommended. Although I am not aware of any upper limit on the number of files that can be in a single directory, it would probably be safest to limit the number of objectIdentifiers within a particular namespace to fewer than 100000. This recommendation can probably be safely ignored by any institution large enough and stable enough to be able to use dynamic file generation from the start and to maintain it for the foreseeable future.

6. Relationship between institution type and file generation method.

In section I.B. four categories of institutions were defined based on the availability of IT resources to that institution. Institutions in categories 1-3 should be able to issue their own guides but availability of IT resources will limit the methods that they can use to generate and deliver the HTML and RDF files associated with their guides.

Because of the availability of server resources and the IT support necessary to implement and maintain them, **category 3** institutions should be able to generate their files dynamically. They should also be able to serve the files needed by the client through content negotiation (the most straightforward method) since their server could be configured to carry out the client-server HTTP dialog and the server-side scripting necessary to select and generate the representation requested (via MIME type) by the client.

Because institutions in **category 2** have sufficient IT support to modify server settings, it would probably be practical for them to implement content negotiation. However, since they would not be running specialized server software, they would probably be using static files, so the method of content negotiation used would have to be one appropriate for static files. The generation of their static files would be accomplished through periodic updating of files associated with records in the database that had changed.

As in the case of category 2 institutions, **category 1** institutions would be limited to the use of static files. However, if their server administrators had limited knowledge or availability (or were non-existent), they might not be able to use content negotiation and be limited to the use of the link method for exposing their RDF files.

In the following sections, I will outline the details of file structure and means of providing appropriate representations for several systems, starting with the most "primitive" (requiring the fewest IT resources). In the examples, I will assume that the guid being resolved is the HTTP URI:

```
http://institution.org/namespace/identifier
```

In all of the examples, HTTP URIs without fragment identifiers are used. However, the methods would apply equally well to hash URIs, since servers strip off (ignore) the fragment identifiers.

It is unlikely that these are the only possible approaches. Better approaches may exist or may be created in the future. As long as those other approaches are RESTful in the sense that they produce the kinds of HTTP and RDF files described here when resolution of a guid is requested, then any approach that works under the circumstances of the data provider is acceptable.

7. Technical details of delivery option 1: Redirection to an HTML file by URL rewrite and access to RDF files by the link method.

This method assumes that static files will be uploaded to the server (via FTP or direct directory access) and that there is a very limited ability to control how the web server responds to URLs.

The first thing that must be determined is how the web server responds to URLs that do not have a file extension. In some cases the web server assumes that all requests for files are for HTML files regardless of the file extension. In that case, the static HTML file should be given the name `identifier` (with no extension) and placed in a directory called `namespace` located below the root directory of the website. In other cases, the web server assumes that a URI without an extension is an unknown file type and sends an HTTP code 404 (Not Found) even if there is an extensionless file in the appropriate directory on the web server. In that case, the only solution is to get the server administrator to apply a rewrite rule that says if a URI is submitted having no file extension, the server should automatically add a ".htm" to the end of the URI which creates the URL of an HTML file. In this case, the static HTML file should be given the name `identifier.htm` and placed in the `namespace` directory as above.

Access to a representation by humans is not a problem since this method directs an html client (i.e. a web browser) to the HTML representation by default. However, since the server is too "stupid" to know what to do if the client requests content type `application/rdf+xml` (and always returns HTML regardless of the request), access to the RDF representation must be done through the link method. Assuming that the RDF file is called `identifier.rdf` and is located in the same directory as the HTML file, the following line should be added to the `<head>` section of the HTML file:

```
<link rel="meta" type="application/rdf+xml" title="RDF" href="identifier.rdf" />
```

This tells a linked data client receiving the HTML file where to look to find the RDF. The `title` attribute can really have any value (although "RDF" is sensible). The `rel` attribute can also probably have other values like "alternate", although I think that meta is the most appropriate (see <http://www.w3.org/TR/html401/struct/links.html> and <http://www.w3.org/QA/Tips/use-links> for reference). The value of the `href` attribute can be either a relative or absolute URI. This means that the RDF metadata could be located at another institution's website if that were beneficial.

It should also be noted that it is not a requirement that the HTML page `identifier.htm` actually produce the human readable version of the metadata related to the resource identified by the guid. It is possible to redirect the client to a different html file that creates the actual web page seen by the user. Information about the guid to be resolved could be passed to the other page through a query string ("`?identifier`" in the examples). An advantage of this is that the human-readable representation for all guides could be displayed by the same html page. The format and appearance of this one page could easily be changed without necessitating changing all of the many individual HTML pages that are served in the process of guid resolution. One method of redirecting to the other page is to use "meta refresh" by including a `<meta>` tag in the head section of the HTML:

```
<meta http-equiv="refresh" content="0;url=http://institution.org/generic.htm?identifier" />
```

This method has been deprecated by W3C and is considered bad form by web designers because in the past it caused problems with a user's web browser Back button. However, all modern web browsers have solved this problem, so I don't see a big problem with using it. Another approach

is to use Javascript to do a page replace. The following code is placed in the <body> section of the HTML:

```
<script type="text/javascript">
window.location.replace("http://institution.org/generic.htm?identifier");
</script>
```

This is considered better form even though this can also cause problems with the user's Back button. However, because the redirect is accomplished by javascript programming, it is possible to create more sophisticated code that does pretty much anything you want to control the behavior of the web browser (see the Web for numerous examples).

If redirection were used to create user-viewed HTML through a single generic page that accepted a query string, that page could utilize the static RDF XML metadata file (through AJAX or perhaps XSLT) as the source of the data used to generate the page. However, the "same origin policy" of Javascript would require that the HTML and RDF files be in the same domain.

Implications. An important implication of this option is that it provides a means for metadata providers with the most meager IT resources to generate guids under their own "brand" (i.e. using their domain name). By using both the link method of redirection to the RDF and one of the HTML redirecting methods, it would be possible for a small institution to completely "outsource" the maintenance of their metadata once the static HTML files were created and put on the server. Because of the use of static files, the metadata would be updated only as often as the database is used to generate new static files.

7. Technical details of delivery option 2: Content negotiation to static RDF and HTML files having the same base URI as the GUID.

Variations on this method is described in the [Content Negotiation page of the Apache website](#). Both variations require that the webserver is running Apache. They also require that the files containing the RDF and HTML representations be present in the directory specified by the HTTP URI of the guid, i.e. `http://institution.org/namespace/` in the example. The names of the RDF and HTML files are formed from the last part of the HTTP URI guid by adding a consistent extension. For RDF files, the extension ".rdf" is recommended (e.g. `identifier.rdf`). For HTML files, the extensions ".htm" or ".html" are recommended (e.g. `identifier.htm` or `identifier.html`). For example:

```
http://institution.org/namespace/identifier.htm
http://institution.org/namespace/identifier.rdf
```

are the URLs for the respective HTML and RDF XML representations of the HTTP URI guid

```
http://institution.org/namespace/identifier
```

There are two methods of accomplishing the actual content negotiation. One method involves placing in the namespace directory a file having the name `identifier.var` which contains information about what files the computer should send to the user depending on the requested content type (`text/html` and `application/rdf+xml`). The server then must

be configured as described in the Apache web page and set up with a rewrite rule that appends ".var" on the end of any URIs that lack a file extension. The contents of the `identifier.var` file for the example would be

```
URI: identifier
```

```
URI: identifier.htm  
content-type: text/html
```

```
URI: identifier.rdf  
content-type: application/rdf+xml
```

The major disadvantage of the ".var" method is that a separate .var file must be created for every guid to be resolved. That increases by 50% the number of static files that have to be kept on the server.

The other method ("MultiViews") involves setting up general rules that apply to all URI resolution requests that don't specify a file extension. In layman's terms the rules are something like "if the user requests an html file, attach '.htm' to the end of the URI, and if the user requests an RDF file, attach '.rdf' to the end of the URI". MultiViews is an option applied to each directory, so if there are multiple directories representing various namespaces, each directory would need to have this option set for it. See the Apache page for details.

Advantages. The Apache content negotiation method provides an RDF file requested by the client directly without requiring the client to search through an HTML file looking for the link to the RDF. The method of content negotiation is very straightforward and conceptually resembles the methods described in <http://www.w3.org/TR/cooluris/#r303uri>.

Disadvantages. The server must be running Apache. The server administrator must be willing and able to make several changes to the server settings. The representation files must be in the directory specified by the HTTP URI and must follow a strict naming convention. If the .var method is used, many additional .var files must be created.

Implications. As was the case for the first delivery option, the guid metadata would only be updated as often as the database was used to create new static files. The same HTML redirection "tricks" used in the first option could be applied here. However, since there is no capability for redirecting to a different directory or domain when a linked data client requests content type `application/rdf+xml`, the RDF files must be on the server that services the domain on which the HTTP URIs are based. This also means that if AJAX acting upon the RDF files is used to generate the HTML representation, the HTML file containing the Javascript would have to be in the same domain as the RDF file. This somewhat limits the ability to "outsource" the creation of the metadata files that represent the guid.

8. Technical details of delivery option 3: Accomplishing content negotiation and file generation dynamically using generic programmable server software.

Since I have not actually used this option and have only limited understanding of server operation, I am indebted to Peter DeVries who shared with me the details of his implementation of content negotiation at <http://about.geospecies.org/>. In this delivery option, the server is run by a web application based on a programmable language such as [Ruby on Rails](#), [IronRuby](#), etc. The exact behavior of the software is determined by the programming, so the implementation could be customized to fit the circumstances of the data provider.

The content negotiation is handled through programming of the web application. Here is an example for Ruby on Rails from Peter:

The "show" controller in `ses_controller.rb`

```
def show
  @se = Se.find_by_se_uid(params[:id]) # Note that this is using the se_uid "v6n7p" as the
  identifier
  if (@se.nil?)
    respond_to do |format|
      format.html {render :template => 'ses/no_se.html'}
      format.rdf {render :template => 'ses/no_se.rdf'}
    end #do
  else
    se_epithet = @se.se_epithet
    se_uid     = @se.se_uid
    se_uuid    = @se.se_uuid

    if params[:format]
      # either the html or rdf representation has been asked for directly, so provide it
      respond_to do |format|
        format.html {render :template => 'ses/show.html'}
        format.rdf {render :template => 'ses/show.rdf'}
        # an alternative here is to call your own method to output the required RDF as a string
        # format.rdf {render :text => my_method_to_make_rdf }
      end #do
    else
      # no format (file extension) specified, so the resource identifier has been requested.
      respond_to will look at HTTP Accept header
      # and do the appropriate redirect
      respond_to do |format|
        format.html {redirect_to :status=>303, :controller=>'ses', :action=>'show',
          :id=>params[:id], :format=>'html'}
        format.rdf {redirect_to :status=>303, :controller=>'ses', :action=>'show',
          :id=>params[:id], :format=>'rdf'}
      end #do
    end #if
  end #if se empty
end
```

Without belaboring the details of the code, it essentially tells the web application that when a client sends an HTTP GET request for a URI that does not specify a file extension, the application should examine the HTTP "Accept header" (in which the client indicates the type of representation it wants) and reply with an HTTP 303 (See Other) response that directs the client to the URI+".html" for HTML requests and the URI+".rdf" for RDF XML. A subsequent request for one or the other file types then results in the application generating the file in the appropriate format using information from a database (rather than sending static files as was the case in the previous two examples) and returning it to the client.

Because the behavior of the web application is programmable, the exact behavior in the content negotiation could be varied. In this code example, the behavior is similar to option 2 in that the URIs of the HTML and RDF representations are related by sharing the same base URI as the guid for the abstract resource. However, with different programming the URIs for the HTML and RDF resources could point anywhere, including to a different domain (although if that domain were not also managed by the same web application there would be little point in doing so). For example

```
http://institution.org/web/namespace/identifier.html  
http://institution.org/rdf/namespace_identifier.rdf
```

could be the URLs for the respective HTML and RDF XML representations of the HTTP URI guid

```
http://institution.org/namespace/identifier
```

The web application could use as a source of metadata any number of database types including but not limited to SQL. Thus if properly coded, the web application could directly access existing databases such as the MySQL database managed by Specify rather than requiring periodic database exports. In this case appropriate security would have to be included in the application design to prevent web clients from accessing personal information or rare taxon occurrence data that might be included in the database.

Advantages. Because the files are generated dynamically from a database, there is no reason why the file URIs would have to be organized hierarchically under different namespaces since there would be no practical restriction on the number of generated files that could share any layer in the hierarchy (including the root of the URI, i.e. the domain name followed by "/"). Thus this option would have the advantage of potentially shorter and simpler HTTP URIs like
`http://institution.org/locallyUniqueIdentifier`

The danger of this approach to URI format would be that if the data provider were forced at some point to drop down to one of the lower-tech delivery options where actual static files were kept in directories, there could be an enormous number of files in the root directory of the website. (See sections II.F.4 and 5)

Disadvantages. It is likely that each implementation would have to be custom-designed to fit the circumstances and database of the data providing institution. However, it might be possible to generate "boilerplate" code that could be adapted to other institutions with minimal modification. The website and guid resolution would probably be more likely to "go down" (resulting in "broken" HTTP URI guids) than the simpler options 1 and 2. The system would probably indefinitely need a trained IT professional to maintain it as well as to make any modifications in the format of the web pages or the RDF properties (metadata terms) included in the metadata. In contrast, the format of web pages in the simpler systems could be adjusted by small changes to the HTML or Javascript in a single generic HTML page file.

Implications. Because this approach is based on programming, there is virtually no limit (other than money and imagination) to how the guids are integrated into database structures and web

environments. However, it is also probably the most "fragile" system and the most susceptible to catastrophe caused by loss of funding.

9. How do we get there from here?

It is my intention that these recommendations be implementable at any time by an institution of nearly any size and resource availability. The Linked Data concept is at its core a distributed system in which the actions of one agent are not dependant on what has been accomplished by someone else. Thus I recommend that institutions at all levels explore the use of HTTP URI guides using a method of file generation and serving that is appropriate for their present circumstances. With careful thought about the format of URIs, an institution should be able to explore the use of HTTP URIs without locking themselves into a particular delivery system. Although in the end I would hope that broad adoption of guides will facilitate the development of a federated SERNEC database, it is not necessary for that to occur in order for institutions to derive the branding and citation benefits that publishing guides will give them. It should also be possible to do this exploration without needing a large funding input.

I recommend the following specific steps be taken:

- To meet the needs of Category 1 institutions, develop stand-alone software that would read in a simple comma delineated file (exported from Excel, Specify, or other database software) that contains a minimal number of required metadata fields, and generate static HTML and RDF files suitable for use with delivery options 1 or 2. The software might be downloadable as an executable application or might be web-based.
- To meet the needs of Category 2 institutions, develop software that would read one or more standard database formats (e.g. MySQL), then generate static files suitable for use with delivery options 1 or 2. The software might use XSLT or other XML based utilities to generate XML that would form the basis of XHTML and RDF/XML used to provide the HTML and RDF representations of the resources identified by the GUIDs.
- To meet the needs of Category 3 institutions, develop template web applications based on Ruby or other programming languages that could access standard database formats and provide dynamic resolution of HTTP URI guides.

Although the methods of generating and delivering the HTML and RDF files will vary, the URI and RDF format recommendations described in sections II.C. and II.E. should be followed to insure interoperability with the broader SERNEC community and flexibility in the event that an institutions's file generation and delivery systems have to change in the future.

If one or more institutions at each level succeed in implementing HTTP URI guides, they can share their experience and computer code with other institutions. At some point, there will be a large enough critical mass of metadata available to proceed with tackling the final challenge discussed in the next section.

III. How will a Linked Data system of biodiversity resources identified by guides be used to do anything useful?

A. The problem of the chicken and the egg.

At the present moment the Semantic Web and Linked Data are to a large extent a pipe dream. There are a few applications such as Google indexing robots and RSS feeds that use aspects of the Semantic Web to do cool things. However, despite years of preaching there are actually very few organizations in the biodiversity world that are using "real" guides that provide RDF metadata to linked data clients. In addition, there are virtually no linked data clients in the biodiversity world that actually do anything useful (at least that I know of).

This state is similar to the early stages of the World Wide Web, where not many people used the Web because there wasn't much content and there wasn't much content because not many people were using the Web. At a certain point, the number of users and amount of content on the Web reached a critical mass where it became "economically viable" to create massive amounts of content. The problem with Web 2.0 (the Semantic Web) is that the technical barriers for creating content are much higher than they were in Web 1.0 when all you really needed to be a web author was a text editor, free FTP software, and a book about HTML.

What I hope to have offered here is a way to lower the technical barriers to implementing guides to the point where at least some people will be able to succeed in using them. If those people can share their experiences, then it will be easier for others to do the same. At some point, there will be enough data providers and sufficient metadata content available for the system within the biodiversity community to be "economically viable". We will be offering "low hanging fruit" that can be plucked by Encyclopedia of Life, Discover Life, Wikipedia, Google, our own SERNEC group, NBII, GBIF, or anybody else who wants to use metadata to create products that they are willing to provide to the public under the licensing conditions similar to [Creative Commons BY-NC-SA](#).

B. How and why do you assemble a database from records that are scattered across the planet?

It is not the purpose of this document to provide details of how to develop a system that makes use of the metadata associated with guides and I'm not qualified to present such details. But I think that a broad outline of how such a system would work is useful for understanding the overall purpose of providing linked data.

1. Provide a site index which links to all resource HTTP URIs within the site.

HTTP URI guides are useless if they cannot be discovered. Guides may be discovered incidentally if they are the object of an RDF property of a subject somewhere else. However, there needs to be a systematic means of discovering all of the guides in a site. A logical means of doing this would be through an RDF Site Summary (RSS, a.k.a. Really Simple Syndication). RSS has

primarily been used as a way to provide news "feeds" from a website, but its defined purpose is to be a document describing URL retrievable items. That seems to be exactly what is needed here. Whether it would be appropriate to describe the guides present on a site using RSS or if it would be better to define a list of guides using some other RDF describable method could be the subject of discussion.

2. Discover many sites that provide RDF metadata for biodiversity resources.

Sites that provide RDF metadata for their guid-identified resources would need to make themselves known to potential data aggregators. One possible venue for this could be through the [Biodiversity Collections Index](#). Another could be through registration with organizations whose purpose is to aggregate biodiversity metadata, such as SERNEC or GBIF.

3. Assemble a RDF triple store.

A database containing data gleaned from RDF sources is called a triple store. In some ways, RDF is similar to other data storage methods in that data can be stored, categorized by terms (or properties), and related to other data. However, RDF differs from other methods of data organization in that it is specifically designed to allow the data user to make inferences that are not explicitly spelled out in the data itself. Two examples of such inferences were given in sections II.E.9. and II.E.10 where images were associated with an individual tree even though the record for the tree itself did not contain a reference to the images and where duplicate specimens were associated with each other even though one herbarium may not have been aware of the existence of the duplicate at the other herbarium. As a federated database grows in size, tasks such as taxonomic revisions become easier as a more complete list of available specimens and images becomes available.

4. Use SPARQL to search the RDF triple store to find useful things.

A query language (SPARQL) has been designed to search RDF data and to create visualizations of related resources. Related resources could be extracted from the database to create range maps, species image comparisons, locality lists, etc.

5. Create a web interface that gives users access

Ultimately we would like to give researchers, educators, and the public access to the resources that we aggregate. This can ultimately be done through a website that generates active content based on interaction between the user and the metadata aggregated on the site.

C. Aside from altruism, is there any benefit for me to do this?

It is fine to talk of the benefits to society at large if the grand Linked Data universe comes into existence. But what direct benefit is there to me or my institution for using HTTP URI guides? What if Web 2.0 never happens?

1. Ability to cite.

A specimen, individual, or image can be unambiguously cited in any publication. In an online publication, the citation itself (in the form of an actionable guid) can form a clickable link that takes the user directly to the metadata (and possibly images) for the resource. In a print publication, a well designed HTTP URI guid is easily written down or typed into a web browser.

2. Branding.

Assigning an HTTP URI guid to a resource permanently associates that resource to your institution because your domain name is a part of the identifier. When a user enters the guid into a web browser, the resulting page can not only tell the user about the resource, but lead the user to other resources offered on your organization's website.

3. Enabling users to find an individual or population.

Since providing location metadata in the form of decimal latitude and longitude for all occurrences is a priority, published guides can provide users a means use a GPS receiver to find the location where the individual was recorded. A link from the HTML metadata representation page to Google Maps can show the user where the individual was located. With a GPS enabled portable device, the user could enter a giud from a brochure, then walk directly to the location of an identified tree in an arboretum, plant in a botanical garden, or significant location on a nature trail. (Of course *dwc:informationWithheld* and *dwc:dataGeneralizations* should be used to protect threatened populations.) A person in an arboretum might come across a tree which is labeled with the QR-Code of the tree's HTTP URI guid, then use a cell phone's camera and an app to access information about the tree through the 3G network.

D. Conclusions

I think it is clear that the benefits of using guides are potentially great. What I hope I have shown is that the cost of creating HTTP URI guides is low enough that real people can actually start using them. If you are waiting for an "official" system for using guides to be imposed upon you from high, you might be waiting for a long time.

The benefit of the Linked Data system is that as long as the basic rules are followed, there is a lot of flexibility in the allowed structure of the guides and the exact metadata terms used to describe resources. In their book [Semantic Web for the Working Ontologist](#), Allemang and Hendler refer the AAA principle: "Anyone can say Anything about Any topic". Obviously, we want to "say things" that others can understand; therefore it is logical that we focus on using Darwin Core, Dublin Core, and MRTG terms. But as long as what we are "saying" about biodiversity resources is written in RDF and makes sense, we really can say just about anything that we want in our metadata and others will be able to understand.

Appendix A - Reference resources

Linked Data, GUIDs, and Semantic Web reference and learning resources

w3schools Online Web tutorials (learn XHTML, XML, Javascript, RDF)

<http://w3schools.com/>

How to Publish Linked Data on the Web

<http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/>

Deploying Linked Data - Part 1: Introduction

http://virtuoso.openlinksw.com/Whitepapers/html/vdld_html/VirtDeployingLinkedDataGuide_Introduction.html#mozTocId762613

Cool URIs for the Semantic Web

<http://www.w3.org/TR/cooluris/>

TDWG GUID/LSID applicability statement

<http://www.tdwg.org/stdtrack/article/download/150/51>

GBIF Persistent-Identifiers statement

<http://www2.gbif.org/Persistent-Identifiers.pdf>

RDF Primer (W3C)

<http://www.w3.org/TR/rdf-primer/>

Baskauf, S.J. 2010. 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* (in press).

<http://bioimages.vanderbilt.edu/pages/baskauf-2010.htm>

Metadata term reference

Dublin Core Terms

<http://dublincore.org/documents/dcmi-terms/>

Darwin Core Terms Quick Reference Guide

<http://dublincore.org/documents/dcmi-terms/>

Media Resources Task Group draft standard v 0.9

http://www.keytonature.eu/wiki/Submission_v0.9

Summary of 2007-08 Discussions of the SERNEC Live Plant Imaging Subgroup

<http://www.sernec.org/?q=node/220>

Follow-up to Summary of SERNEC Live Plant Imaging subgroup discussion 9 May 2010

<http://www.sernec.org/?q=node/234>

XML, RDF, and Linked Data creation and validation tools (all free)

jEdit text editor (automatically checks character encoding and XML well-formedness if XML plugin is installed)

<http://www.jedit.org/>

WinSCP FTP (file uploading) and SFTP (secure FTP) tool

<http://winscp.net/>

XML validator (includes checking against schemas pointed to within document)

<http://www.validome.org/xml/validate/>

rdf:about RDF Validator and Converter

<http://www.rdfabout.com/demo/validator/>

W3C RDF Validation Service

<http://www.w3.org/RDF/Validator/>

Vapour Linked Data validator

<http://vapour.sourceforge.net/>

OpenLink RDF Browser

<http://demo.openlinksw.com/rdfbrowser/>

Functioning examples of biodiversity-related websites that have implemented GUIDs with content negotiation

Biodiversity Collections Index

<http://www.biodiversitycollectionsindex.org/>

GeoSpecies Knowledge Base

<http://about.geospecies.org/>

Bioimages

<http://bioimages.vanderbilt.edu/>

Appendix B - RDF example for a specimen, its individual, and its image in a single file

Notes:

- This file can be downloaded from <http://bioimages.vanderbilt.edu/rdf/examples/hb123456.rdf> .
- Many of the URIs in this example are fictional, so don't expect the file to be functional.
- In this example the minimal metadata are provided. Additional properties for the specimen could be provided using other Darwin Core terms as desired.
- An effort was made to define properties using multiple vocabularies to allow the metadata to be understood by the widest range of linked data clients (e.g. *foaf:depiction* and *dwc:associatedMedia* to refer to the image of the specimen).

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:dwc="http://rs.tdwg.org/dwc/terms/"
  xmlns:xmp="http://ns.adobe.com/xap/1.0/"
  xmlns:xmpRights="http://ns.adobe.com/xap/1.0/rights/"
  xmlns:Iptc4xmpExt="http://iptc.org/std/Iptc4xmpExt/2008-02-29/"
  xmlns:mbank="http://www.morphbank.net/schema/morphbank#"
  xmlns:mix="http://www.loc.gov/mix/v20"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:mrtg="http://xxx.org/XXX/"
  xmlns:sernec="http://bioimages.vanderbilt.edu/rdf/terms#"
  >
  <!-- Note: as of 2010-05-18 a namespace for the MRTG schema had not been declared-->

  <rdf:Description rdf:about="http://herbarium.org/hb123456#ind">
    <mrtg:MetadataLanguage>en</mrtg:MetadataLanguage>
    <!--
      Basic information about the individual
    -->

    <dcterms:description>Field individual of Quercus alba</dcterms:description>
    <!-- Currently there is no Darwin Core class for individuals that can be used as
      a value for rdfs:type. As a temporary measure, I defined a class for individuals
      and used that class to type the individuals here.-->
    <rdfs:type rdf:resource="http://bioimages.vanderbilt.edu/rdf/terms#Individual"/>
    <dcterms:type rdf:resource="http://purl.org/dc/dcmitype/PhysicalObject"/>
    <dwc:establishmentMeans>native</dwc:establishmentMeans>
    <!--
      Relationships of the individual to other resources
    -->

    <sernec:derivativeOccurrence rdf:resource="http://herbarium.org/hb123456"/>
    <!-- Determinations applied to the individual-->
    <dwc:identificationID rdf:resource="http://herbarium.org/hb123456#19290"/>
  </rdf:Description>

  <!-- Note: fragment identifier formed from ITIS TSN for Quercus alba -->
  <rdf:Description rdf:about="http://herbarium.org/hb123456#19290" >
    <mrtg:MetadataLanguage>en</mrtg:MetadataLanguage>
    <!--
      Basic information about the determination
    -->

    <dcterms:description>Determination of Quercus alba for the individual
    http://herbarium.org/hb123456#ind</dcterms:description>
    <rdfs:type rdf:resource="http://rs.tdwg.org/dwc/terms/Identification" />
    <dwc:identifiedBy rdf:resource="http://herbarium.org/people/jane-curator#person"/>
    <dwc:dateIdentified>1997-06-23</dwc:dateIdentified>
    <!--
```

```

Relationship of the determination to other resources
-->
<sernec:identifiesIndividual rdf:resource="http://herbarium.org/hb123456#ind"/>
<sernec:basedOnOccurrence rdf:resource="http://herbarium.org/hb123456"/>
<dwc:taxonConceptID rdf:resource="http://lod.geospecies.org/ses/waK4b"/>
<!--
Direct literals for the determination can be found here without resolving the taxonConceptID
-->
<dwc:family>Fagaceae</dwc:family>
<dwc:genus>Quercus</dwc:genus>
<dwc:specificEpithet>alba</dwc:specificEpithet>
<dwc:taxonRank>species</dwc:taxonRank>
<dwc:scientificNameAuthorship>L.</dwc:scientificNameAuthorship>
</rdf:Description>

<rdf:Description rdf:about="http://herbarium.org/hb123456">
  <mrtg:MetadataLanguage>en</mrtg:MetadataLanguage>
  <!--
Basic information about the specimen
-->
  <dcterms:description>Preserved specimen of Quercus alba</dcterms:description>
  <dcterms:identifier>http://herbarium.org/hb123456</dcterms:identifier>
  <dcterms:creator>National Herbarium of Colaxico</dcterms:creator>
  <dcterms:created>1997-06-23</dcterms:created>
  <rdfs:type rdf:resource="http://rs.tdwg.org/dwc/terms/Occurrence"/>
  <dcterms:type rdf:resource="http://purl.org/dc/dcmitype/PhysicalObject"/>
  <dwc:basisOfRecord rdf:resource="http://rs.tdwg.org/dwc/dwctype/PreservedSpecimen"/>
  <!-- Note: establishmentMeans listed here for clients that don't "understand" individuals -->
  <dwc:establishmentMeans>native</dwc:establishmentMeans>
  <sernec:documentsDistribution>true</sernec:documentsDistribution>
  <dwc:recordedBy rdf:resource="http://herbarium.org/people/jane-curator#person"/>
  <dwc:eventDate>1997-06-23</dwc:eventDate>
  <dwc:collectionID rdf:resource="http://biocol.org/urn:lsid:biocol.org:col:92134"/>
  <!-- Minimal taxonomic data given here for clients too "dumb" to figure out how the
specimen is related to its determination
-->
  <dwc:family>Fagaceae</dwc:family>
  <dwc:genus>Quercus</dwc:genus>
  <dwc:specificEpithet>alba</dwc:specificEpithet>
  <!--
Relationships of the specimen to other resources
-->
  <foaf:isPrimaryTopicOf rdf:resource="http://herbarium.org/hb123456.rdf"/>
  <foaf:isPrimaryTopicOf rdf:resource="http://herbarium.org/hb123456.htm"/>
  <dwc:individualID rdf:resource="http://herbarium.org/hb123456#ind"/>
  <!-- Note: identification listed here for clients that don't "understand" individuals -->
  <dwc:identificationID rdf:resource="http://herbarium.org/hb123456#19290"/>
  <sernec:derivedFrom rdf:resource="http://herbarium.org/hb123456#ind"/>
  <sernec:derivativeOccurrence rdf:resource="http://herbarium.org/hb123456#img"/>
  <dwc:associatedMedia rdf:resource="http://herbarium.org/hb123456#img"/>
  <foaf:depiction rdf:resource="http://herbarium.org/hb123456#img"/>
  <sernec:usedInDetermination rdf:resource="http://herbarium.org/hb123456#19290"/>
  <!--
Location information
-->
  <dwc:decimalLatitude>36.38356</dwc:decimalLatitude>
  <dwc:decimalLongitude>-87.00681</dwc:decimalLongitude>
  <dwc:geodeticDatum>epsg:4326</dwc:geodeticDatum>
  <dwc:coordinateUncertaintyInMeters>10</dwc:coordinateUncertaintyInMeters>
  <!-- Additional Darwin Core location terms (dwc:county, dwc:stateProvince, etc.) can be listed -->
  <dwc:locality>US 41A 1.6 mi. SE of TN 49</dwc:locality>
  <!-- Note: the "true" value of sernec:documentsDistribution means that the creation of this
occurrence (the specimen) serves as evidence that a taxon representative exists in the wild at the
location given. -->
</rdf:Description>

<rdf:Description rdf:about="http://herbarium.org/hb123456#img">
  <mrtg:MetadataLanguage>en</mrtg:MetadataLanguage>
  <dcterms:identifier>http://herbarium.org/hb123456#img</dcterms:identifier>
  <dcterms:creator>National Herbarium of Colaxico</dcterms:creator>
  <dcterms:created>2006-09-05</dcterms:created>

```

```

    <rdfs:type rdf:resource="http://rs.tdwg.org/dwc/terms/Occurrence"/>
    <dcterms:type rdf:resource="http://purl.org/dc/dcmitype/StillImage"/>
    <!--
DigitalStillImage does not have a normative URI because it is not (yet) an accepted Dwc type
-->
    <dwc:basisOfRecord>DigitalStillImage</dwc:basisOfRecord>
    <sernec:documentsDistribution>false</sernec:documentsDistribution>
    <!-- Note: the "false" value of sernec:documentsDistribution means that the creation of this
occurrence (the image) does not serve as evidence that a taxon representative exists in the wild
at the National Herbarium of Colaxico). -->
    <dwc:collectionID rdf:resource="http://biocol.org/urn:lsid:biocol.org:col:92134"/>
    <!--
Other properties of the image related to intellectual property rights, use and attribution guidelines, etc.
-->
    <dcterms:rights>(c) 2006 National Herbarium of Colaxico</dcterms:rights>
    <xmpRights:owner>National Herbarium of Colaxico</xmpRights:owner>
    <Iptc4xmpExt:creditLine>National Herbarium of Colaxico
http://herbarium.org/</Iptc4xmpExt:creditLine>
    <mbank:view>77407</mbank:view>
    <!-- Standard views not yet established for herbarium specimens -->
    <Iptc4xmpExt:CVterm rdf:resource="http://bioimages.vanderbilt.edu/rdf/stdview#000000"/>
    <dwc:title>Quercus alba (Fagaceae) specimen</dwc:title>
    <xmpRights:UsageTerms>Available under Creative Commons Attribution-Noncommercial-Share Alike 3.0
license</xmpRights:UsageTerms>
    <xmpRights:WebStatement>http://creativecommons.org/licenses/by-nc-
sa/3.0/us/</xmpRights:WebStatement>
    <dcterms:description>Image of a Quercus alba (Fagaceae) specimen</dcterms:description>
    <mrtg:attributionLinkURL>http://herbarium.org/contact.htm</mrtg:attributionLinkURL>
    <mrtg:attributionLogoURL>http://herbarium.org/logo.jpg</mrtg:attributionLogoURL>
    <!-- Image collection status = 0 because it's not a live plant-->
    <sernec:sernecImageCollectionStatus>0</sernec:sernecImageCollectionStatus>
    <xmp:rating>5</xmp:rating>
    <!--
Relationships of the image to other resources.
-->
    <dwc:individualID rdf:resource="http://herbarium.org/hb123456#ind"/>
    <sernec:derivedFrom rdf:resource="http://herbarium.org/hb123456"/>
    <foaf:depicts rdf:resource="http://herbarium.org/hb123456"/>
    <mrtg:hasServiceAccessPoint rdf:resource="http://herbarium.org/hb123456#bq"/>
    <!-- Note: additional service access point references can be listed here -->
</rdf:Description>

    <mrtg:hasServiceAccessPoint rdf:about="http://herbarium.org/hb123456#bq">
    <mrtg:variant>Best Quality</mrtg:variant>
    <mrtg:accessURL>http://herbarium.org/images/dsc55794.jpg</mrtg:accessURL>
    <dcterms:format>image/jpeg</dcterms:format>
    <mix:imageWidth>3000</mix:imageWidth>
    <mix:imageHeight>3582</mix:imageHeight>
    <mix:xSamplingFrequency>87</mix:xSamplingFrequency>
    <mix:ySamplingFrequency>87</mix:ySamplingFrequency>
    <mix:samplingFrequencyUnit>cm</mix:samplingFrequencyUnit>
    </mrtg:hasServiceAccessPoint>
    <!-- Note: additional service access points of different qualities can be listed here -->

    <!--
Information about the metadata itself
-->
    <rdf:Description rdf:about="http://herbarium.org/hb123456.rdf">
    <dcterms:description>RDF formatted description of the preserved specimen
http://herbarium.org/hb123456</dcterms:description>
    <dcterms:creator>National Herbarium of Colaxico</dcterms:creator>
    <dcterms:created>2006-09-08T12:01:30-0800</dcterms:created>
    <dcterms:language>en</dcterms:language>
    <dcterms:modified>2009-10-07T09:14:08-0800</dcterms:modified>
    <xmp:MetadataDate>2009-10-07T09:14:08-0800</xmp:MetadataDate>
    <dcterms:references rdf:resource="http://herbarium.org/hb123456"/>
    <foaf:primaryTopic rdf:resource="http://herbarium.org/hb123456"/>
    </rdf:Description>
</rdf:RDF>

```

Appendix C - RDF example of metadata for an individual in a separate file

Notes:

- This file can be downloaded from <http://bioimages.vanderbilt.edu/ind-baskauf/66920.rdf>.
- It is a functional file and the URIs should be valid and surfable in a linked data client like the OpenLink RDF Browser (<http://demo.openlinksw.com/rdfbrowser/>)
- An effort was made to define properties using multiple vocabularies to allow the metadata to be understood by the widest range of linked data clients (e.g. *foaf:depiction* and *dwc:associatedMedia* to refer to the live plant image of the tree).
- Given that this is a real "live" metadata file, the version on the Internet is subject to change.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:dwc="http://rs.tdwg.org/dwc/terms/"
  xmlns:sernec="http://bioimages.vanderbilt.edu/rdf/terms#"
  xmlns:mrtg="http://xxx.org/XXX/"
  xmlns:xmp="http://ns.adobe.com/xap/1.0/"
  xmlns:bibo="http://purl.org/ontology/bibo/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  >
<rdf:Description rdf:about="http://bioimages.vanderbilt.edu/ind-baskauf/66920">
  <mrtg:MetadataLanguage>en</mrtg:MetadataLanguage>
  <!--Basic information about the individual-->
  <dcterms:identifier>http://bioimages.vanderbilt.edu/ind-baskauf/66920</dcterms:identifier>
  <dcterms:description>Field individual of Quercus lobata Nee with GUID:
http://bioimages.vanderbilt.edu/ind-baskauf/66920</dcterms:description>
  <!-- Currently there is no Darwin Core class for individuals that can be used as
a value for rdfs:type. As a temporary measure, I defined a class for individuals
and used that class to type the individuals here.-->
  <rdfs:type rdf:resource="http://bioimages.vanderbilt.edu/rdf/terms#Individual" />
  <dcterms:type rdf:resource="http://purl.org/dc/dcmitype/PhysicalObject" />
  <dwc:establishmentMeans>native</dwc:establishmentMeans>

  <!-- Relationships of the individual to other resources. -->
  <foaf:isPrimaryTopicOf rdf:resource="http://bioimages.vanderbilt.edu/ind-baskauf/66920.rdf" />
  <foaf:isPrimaryTopicOf rdf:resource="http://bioimages.vanderbilt.edu/ind-baskauf/66920.htm" />
  <bibo:Webpage rdf:resource="http://bioimages.vanderbilt.edu/ind-baskauf/66920.htm" />
  <!-- Images that are derived from the individual -->
  <sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66924"/>
  <dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66924"/>
  <foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66924"/>

  <sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66925"/>
  <dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66925"/>
  <foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66925"/>

  <sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66926"/>
  <dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66926"/>
  <foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66926"/>

  <sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66927"/>
  <dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66927"/>
  <foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66927"/>

  <sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66928"/>
  <dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66928"/>
```

```

<foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66928"/>

<sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66929"/>
<dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66929"/>
<foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66929"/>

<sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66930"/>
<dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66930"/>
<foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66930"/>

<sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66934"/>
<dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66934"/>
<foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66934"/>

<sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66935"/>
<dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66935"/>
<foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66935"/>

<sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66937"/>
<dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66937"/>
<foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66937"/>

<sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66938"/>
<dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66938"/>
<foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66938"/>

<sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66939"/>
<dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66939"/>
<foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66939"/>

<sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66940"/>
<dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66940"/>
<foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66940"/>

<sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66941"/>
<dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66941"/>
<foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66941"/>

<sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66942"/>
<dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66942"/>
<foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66942"/>

<sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66943"/>
<dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66943"/>
<foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66943"/>

<sernec:derivativeOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66921"/>
<dwc:associatedMedia rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66921"/>
<foaf:depiction rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66921"/>

<!-- Determinations applied to the individual-->
<dwc:identificationID rdf:resource="http://bioimages.vanderbilt.edu/ind-baskauf/66920#19370" />
</rdf:Description>

<rdf:Description rdf:about="http://bioimages.vanderbilt.edu/ind-baskauf/66920#19370" >
  <dcterms:description>Determination of Quercus lobata Nee for the individual
  http://bioimages.vanderbilt.edu/ind-baskauf/66920</dcterms:description>
  <rdfs:type rdf:resource="http://rs.tdwg.org/dwc/terms/Identification" />
  <dwc:identifiedBy rdf:resource="http://people.vanderbilt.edu/~steve.baskauf/foaf.rdf#me" />
  <dwc:dateIdentified>2008-07-20</dwc:dateIdentified>

  <!-- Relationship of the determination to other resources -->
  <sernec:identifiesIndividual rdf:resource="http://bioimages.vanderbilt.edu/ind-baskauf/66920"/>
  <!-- <sernec:basedOnOccurrence> and <dwc:taxonConceptID> are not yet implemented
  <sernec:basedOnOccurrence rdf:resource="http://bioimages.vanderbilt.edu/baskauf/[some-image-number]"/>
  dwc:taxonConceptID rdf:resource="http://lod.geospecies.org/ses/[some-identifier]"/>
  -->

  <!-- In lieu of a functional taxonConceptID, the taxonomic information will be expressed as literals-->
  <dwc:class>Magnoliopsida</dwc:class>
  <dwc:order>Fagales</dwc:order>

```

```

    <dwc:family>Fagaceae</dwc:family>
    <dwc:genus>Quercus</dwc:genus>
    <dwc:specificEpithet>lobata</dwc:specificEpithet>
    <dwc:taxonRank>species</dwc:taxonRank>
    <dwc:scientificNameAuthorship>Nee</dwc:scientificNameAuthorship>
</rdf:Description>

<!--
Information about the metadata document itself
-->
<rdf:Description rdf:about="http://bioimages.vanderbilt.edu/ind-baskauf/66920.rdf">
    <dcterms:identifier>http://bioimages.vanderbilt.edu/ind-baskauf/66920.rdf</dcterms:identifier>
    <dcterms:description>RDF formatted description of the living organism
http://bioimages.vanderbilt.edu/ind-baskauf/66920</dcterms:description>
    <dcterms:creator rdf:resource="http://biocol.org/urn:lsid:biocol.org:col:35115"/>
    <dcterms:language>en</dcterms:language>
    <dcterms:modified>2010-03-08</dcterms:modified>
    <xmp:MetadataDate>2010-03-08</xmp:MetadataDate>
    <dcterms:references rdf:resource="http://bioimages.vanderbilt.edu/ind-baskauf/66920"/>
    <foaf:primaryTopic rdf:resource="http://bioimages.vanderbilt.edu/ind-baskauf/66920"/>
</rdf:Description>
</rdf:RDF>

```


Appendix D - RDF example of metadata for a live plant image in a separate file

Notes:

- This file can be downloaded from <http://bioimages.vanderbilt.edu/baskauf/66921.rdf>.
- It is a functional file and the URIs should be valid and surfable in a linked data client like the OpenLink RDF Browser (<http://demo.openlinksw.com/rdfbrowser/>)
- As with the example of Appendix C, the version of this file on the Internet is subject to change over time.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:dwc="http://rs.tdwg.org/dwc/terms/"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:sernec="http://bioimages.vanderbilt.edu/rdf/terms#"
  xmlns:mrtg="http://xxx.org/XXX/"
  xmlns:xmp="http://ns.adobe.com/xap/1.0/"
  xmlns:xmpRights="http://ns.adobe.com/xap/1.0/rights/"
  xmlns:Iptc4xmpExt="http://iptc.org/std/Iptc4xmpExt/2008-02-29/"
  xmlns:mbank="http://www.morphbank.net/schema/morphbank#"
  xmlns:mix="http://www.loc.gov/mix/v20"
  xmlns:bibo="http://purl.org/ontology/bibo/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  >
<rdf:Description rdf:about="http://bioimages.vanderbilt.edu/baskauf/66921">
  <mrtg:MetadataLanguage>en</mrtg:MetadataLanguage>
  <!--Basic information about the image-->
  <dcterms:identifier>http://bioimages.vanderbilt.edu/baskauf/66921</dcterms:identifier>
  <dwc:creator rdf:resource="http://people.vanderbilt.edu/~steve.baskauf/foaf.rdf#me" />
  <dcterms:created>2008-07-20T16:10:27</dcterms:created>
  <rdfs:type rdf:resource="http://rs.tdwg.org/dwc/terms/Occurrence" />
  <dcterms:type rdf:resource="http://purl.org/dc/dcmitype/StillImage" />
  <!-- DigitalStillImage does not have a normative URI because it is not (yet) an accepted DwC type -->
  <dwc:basisOfRecord>DigitalStillImage</dwc:basisOfRecord>
  <dwc:occurrenceRemarks>test remark</dwc:occurrenceRemarks>
  <sernec:documentsDistribution>true</sernec:documentsDistribution>
  <dwc:recordedBy rdf:resource="http://people.vanderbilt.edu/~steve.baskauf/foaf.rdf#me" />
  <dwc:eventDate>2008-07-20T16:10:27</dwc:eventDate>
  <dwc:collectionID rdf:resource="http://biocol.org/urn:lsid:biocol.org:col:35115" />
  <dwc:institutionCode>bioimages</dwc:institutionCode>
  <dwc:collectionCode>baskauf</dwc:collectionCode>
  <dwc:catalogNumber>66921</dwc:catalogNumber>
  <!--Other properties of the image related to intellectual property rights, use and attribution
  guidelines, etc. -->
  <dcterms:rights>(c) 2010 Steven J. Baskauf</dcterms:rights>
  <xmpRights:owner>Steven J. Baskauf</xmpRights:owner>
  <Iptc4xmpExt:creditLine>Steven J. Baskauf http://bioimages.vanderbilt.edu/</Iptc4xmpExt:creditLine>
  <mbank:view>463267</mbank:view>
  <Iptc4xmpExt:CVterm rdf:resource="http://bioimages.vanderbilt.edu/rdf/stdview#010101" />
  <dwc:title>Quercus lobata (Fagaceae) - whole tree (or vine) - general</dwc:title>
  <xmpRights:UsageTerms>Available under Creative Commons Attribution-Noncommercial-Share Alike 3.0
  license</xmpRights:UsageTerms>
  <xmpRights:WebStatement>http://creativecommons.org/licenses/by-nc-sa/3.0/us/</xmpRights:WebStatement>
  <dcterms:description>Image of Quercus lobata (Fagaceae) - whole tree (or vine) -
  general</dcterms:description>
  <mrtg:attributionLinkURL>http://bioimages.vanderbilt.edu/contact/baskauf.htm</mrtg:attributionLinkURL>
  <mrtg:attributionLogoURL>http://bioimages.vanderbilt.edu/contact/baskauf-logo</mrtg:attributionLogoURL>
  <sernec:sernecImageCollectionStatus>2</sernec:sernecImageCollectionStatus>
  <xmp:rating>5</xmp:rating>

  <!-- Relationships of the image to other resources. -->
```

```

<foaf:isPrimaryTopicOf rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66921.htm" />
<foaf:isPrimaryTopicOf rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66921.rdf" />
<bibo:Webpage rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66921.htm" />
<sernec:derivedFrom rdf:resource="http://bioimages.vanderbilt.edu/ind-baskauf/66920" />
<foaf:depicts rdf:resource="http://bioimages.vanderbilt.edu/ind-baskauf/66920" />
<dwc:individualID rdf:resource="http://bioimages.vanderbilt.edu/ind-baskauf/66920" />

<!-- Not yet implemented
<sernec:usedInDetermination rdf:resource="http://bioimages.vanderbilt.edu/ind-baskauf/66920#[TSNID]"/>
-->
<!--
Location information
-->
<dwc:decimalLatitude>37.92019</dwc:decimalLatitude>
<dwc:decimalLongitude>-121.9419</dwc:decimalLongitude>
<dwc:geodeticDatum>epsg:4326</dwc:geodeticDatum>
<dwc:coordinateUncertaintyInMeters>10</dwc:coordinateUncertaintyInMeters>
<dwc:locality>Mitchell Canyon, Mt. Diablo State Park</dwc:locality>
<dwc:minimumElevationInMeters>171</dwc:minimumElevationInMeters>
<dwc:maximumElevationInMeters>171</dwc:maximumElevationInMeters>
<dwc:continent>NA</dwc:continent>
<dwc:countryCode>US</dwc:countryCode>
<dwc:stateProvince>California</dwc:stateProvince>
<dwc:county>Contra Costa</dwc:county>
<!--
Links to ServiceAccessPoints
-->
<mrtg:hasServiceAccessPoint rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66921#bq" />
<mrtg:hasServiceAccessPoint rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66921#tn" />
<mrtg:hasServiceAccessPoint rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66921#lq" />
<mrtg:hasServiceAccessPoint rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66921#gq" />
</rdf:Description>

<!--
ServiceAccessPoints provide information about alternative versions of the image having different resolutions
-->
<mrtg:hasServiceAccessPoint rdf:about="http://bioimages.vanderbilt.edu/baskauf/66921#bq">
  <mrtg:variant>Best Quality</mrtg:variant>
  <mrtg:accessURL> http://bioimages.vanderbilt.edu/baskauf/66921.jpg </mrtg:accessURL>
  <dcterms:format>image/jpeg</dcterms:format>
  <mix:imageWidth>3456</mix:imageWidth>
  <mix:imageHeight>2304</mix:imageHeight>
</mrtg:hasServiceAccessPoint>
<mrtg:hasServiceAccessPoint rdf:about="http://bioimages.vanderbilt.edu/baskauf/66921#tn">
  <mrtg:variant>Thumbnail</mrtg:variant>
  <mrtg:accessURL>http://bioimages.vanderbilt.edu/tn/baskauf/t66921.jpg</mrtg:accessURL>
  <dcterms:format>image/jpeg</dcterms:format>
  <mix:imageWidth>100</mix:imageWidth>
  <mix:imageHeight>67</mix:imageHeight>
</mrtg:hasServiceAccessPoint>
<mrtg:hasServiceAccessPoint rdf:about="http://bioimages.vanderbilt.edu/baskauf/66921#lq">
  <mrtg:variant>Lower Quality</mrtg:variant>
  <mrtg:accessURL>http://bioimages.vanderbilt.edu/lq/baskauf/w66921.jpg</mrtg:accessURL>
  <dcterms:format>image/jpeg</dcterms:format>
  <mix:imageWidth>480</mix:imageWidth>
  <mix:imageHeight>320</mix:imageHeight>
</mrtg:hasServiceAccessPoint>
<mrtg:hasServiceAccessPoint rdf:about="http://bioimages.vanderbilt.edu/baskauf/66921#gq">
  <mrtg:variant>Good Quality</mrtg:variant>
  <mrtg:accessURL>http://bioimages.vanderbilt.edu/gq/baskauf/g66921.jpg</mrtg:accessURL>
  <dcterms:format>image/jpeg</dcterms:format>
  <mix:imageWidth>1024</mix:imageWidth>
  <mix:imageHeight>683</mix:imageHeight>
</mrtg:hasServiceAccessPoint>

<!--
Information about the metadata document itself
-->
<rdf:Description rdf:about="http://bioimages.vanderbilt.edu/baskauf/66921.rdf">
  <dcterms:identifier>http://bioimages.vanderbilt.edu/baskauf/66921.rdf</dcterms:identifier>

```

```
<dcterms:description>RDF formatted description of the live organism image
http://bioimages.vanderbilt.edu/baskauf/66921</dcterms:description>
<dcterms:creator rdf:resource="http://biocol.org/urn:lsid:biocol.org:col:35115"/>
<dcterms:language>en</dcterms:language>
<dcterms:modified>2010-04-14T21:27:58</dcterms:modified>
<xmp:MetadataDate>2010-04-14T21:27:58</xmp:MetadataDate>
<dcterms:references rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66921"/>
<foaf:primaryTopic rdf:resource="http://bioimages.vanderbilt.edu/baskauf/66921"/>
</rdf:Description>
</rdf:RDF>
```