Annotation-based enrichment of Digital Objects using open-source frameworks

Name of Authors (alphabetically) - Marcus Emmanuel Barnes, Natkeeran Ledchumykanthan, Kim Pham, Kirsta Stapelfeldt

Abstract

The W3C Web Annotation Data Model, Protocol, and Vocabulary unify approaches to annotations across the web, enabling their aggregation, discovery and persistence over time. In addition, new javascript libraries provide the ability for users to annotate multi-format content. In this paper, we describe how we have leveraged these developments to provide annotation features alongside Islandora’s existing preservation, access, and management capabilities. We also discuss our experience developing with the Web Annotation Model as an open web architecture standard, as well as our approach to integrating mature external annotation libraries. The resulting software (the Web Annotation Utility Module for Islandora) accommodates annotation across multiple formats. This solution can be used in various Digital Scholarship contexts.

Introduction

Annotation has been a topic of interest since the advent of digital formats and computer networking, but the conversation moved toward productive solutions at a symposium on "Humanities Computing: formal methods, experimental practice" sponsored by King's College, London on May 13, 2000. At this symposium, John Unsworth coined the phrase "scholarly primitives" to describe several activities (including annotation) that cross disciplinary boundaries and are essential to most academic endeavors[1]. Since his talk, the frequency with which the term "scholarly primitive" has been used[2] is a reflection of the attractiveness of thinking about annotation as something that is disciplinary-agnostic and discrete from other web-based creation and publication activities.

Since that time, great strides have been made in the ability to provide web-based annotation features. It has become possible to annotate digital files on the web, persist annotations over time, group them over copies of a resource, and discover annotations that are related by subject or content. The ability to think of annotation as a unique activity has arguably been important to these developments. However, in describing what is different about annotation in comparison to other digital objects, we might forget what is the same. Annotations are digital objects or entities themselves and can benefit from the same tools and procedures. Indeed, at the same moment
that Unsworth coined the term “digital primitives,” he acknowledged that considering annotation as axiomatic was philosophically problematic[1].

With this in mind, our team at the University of Toronto Scarborough Library leveraged existing work to develop a Web Annotation Utility[3] that operates in the Islandora Framework. We added features to the existing infrastructure for digital objects to enable the creation of annotations and associations/relations with other digital objects. W3C’s new proposed recommendations for a Web Annotation Model[4], Protocol[5], and Vocabulary[6] and the maturation of annotation libraries provide new opportunities for web annotation features in a digital repository.

A Web Annotation Utility in Islandora’s Ecosystem

Islandora[7] is an open-source framework supporting digital repositories and collections. The framework has two major components: Drupal[8] (a popular content management system) and Fedora[9] (a best-practices digital repository). Islandora developers provide Drupal modules that, when installed with their dependencies, allow users to manage, create, and access Fedora repository items (digital objects comprised of files, or datastreams) in a Drupal interface. Core indexing functions are provided through Islandora-specific modules that provide custom integration with Apache Solr [10]. Outside of Islandora’s core indexing and Fedora connection functions, modules fall broadly into three categories:

- **Solution Packs** - package custom forms, workflows, and content models. Content models are templates for an object of a particular type. Solution packs usually service a data format (such as newspapers or large images)
- **Utility modules** - provide functions to some or all repository objects (such as workflow or the creation of checksums)
- **Viewers** - provide customization of the presentation layer specific to object content-types. Islandora viewer modules are part of the “view” layer, used in Islandora’s multi-tier architecture. (These are players for things like video or large image content)

The DSU’s Web Annotation Utility Module falls into the Utility module category. The module installs content models that follow the W3C Web Annotation Data Model, Protocol, and Vocabulary, leverages existing viewers, and implements javascript libraries to form a sustainable base for web annotation of multimedia content. The module allows users to annotate large and basic images. It also provides the ability to annotate video, audio, and oral history objects. All annotations, regardless of what type of content they are written for, adhere to a common data model. On the front end, popular viewer-specific libraries are used to provide users with the ability to create annotations.

We hope our approach will facilitate migration to the next major version of Fedora (Fedora 4) which, as a native linked-data platform, will allow for greater adherence to the W3C Web
Annotation Data Model, Protocol, and Vocabulary. We also hope that the Web Annotation Utility Module provides an approach that could work for other existing or emerging Islandora viewers. For example, this approach could be extended to provide annotations on other types of digital objects, such as PDFs.

This is not the first Islandora annotation module. An Islandora Image Annotation Solution Pack[11] was developed for use in Islandora by the University of Prince Edward Island's Robertson Library. The initial module was based on the Open Annotation Collaboration (OAC) standard[12] a precursor to the current W3C Web Annotation Model, Protocol, and Vocabulary[13]. Our Web Annotation Utility Module does not adopt the same approach or use the same data model as the Islandora Image Annotation Solution Pack, which is currently hosted in Islandora Labs[14].

The content models used by the Web Annotation Utility Module initially appeared in an early version of the Islandora Oral History Solution Pack[15]. However, we realized that our users sought to perform annotations on other types of objects and decided to repurpose this functionality and create this utility module. Our desire for a more agnostic approach (as well as the desire to make annotations separate Islandora objects) also prompted us to move away from the image-specific framework of the Islandora Image Annotation Solution Pack.

The W3C Recommendations for Annotations

On January 17 2017, the W3C Web Annotation Data Model, Protocol, and Vocabulary documents became proposed recommendations. The documents focus on reuse of annotations across systems by describing a simple format (in JSON-LD) that can be used in multiple contexts. The Web Annotation Data Model describes annotations "to be a set of connected resources, typically including a body and target [...] that [...] is related to the target"[4]. Annotations have at least one target, and may also have a body (see Figure 1). The annotation and target each contain metadata describing their properties and relationships. The body may be embedded in the annotation or consist of another digital object. The recommendation describes the example of a post made about a particular web page. In this example, the "annotation" would link together the post as the body and the web page as the target. In contrast, a comment on a webpage could be stored more simply as an annotation containing the body of the comment.

![Diagram of annotation components](image)

**Figure 1** - From the Web Annotation Data Model document [4], depicts the conceptual separation of annotation from the body and the target of the annotation.
The second recommendation, the *Web Annotation Protocol* [5], describes the standard for interactions between clients and servers via REST architecture. The *Protocol* prescribes that annotations are to be managed via an Annotation Container (which conforms to the Linked Data Platform specification [16] of an LDP Container). The *Protocol* also describes CRUD operations for annotation, retrieval and discovery mechanisms via the container (AnnotationContainer), and paging mechanisms for breaking down the response (AnnotationPage). The *Protocol* assumes a Linked Data Platform and does not specify how containers are created or deleted, or how to handle AnnotationContainer if an LDP is not being used. In Fedora 3, the *Web Annotation Utility Module* implements the AnnotationContainer and Annotation using separate content models (templates for digital objects, as described above).

The final recommendation, the *Web Annotation Vocabulary* [6], defines a vocabulary for annotations. Our module adheres to this vocabulary in the content of AnnotationContainer and Annotation objects. These content models are described in more detail in the following section.

### Web Annotation Utility Module Content Models and Workflow

The content models installed by the *Web Annotation Utility Module* are the Islandora Web Annotation Content Model (islandora:WADMCModel) and the Islandora Web Annotation Container Content Model (islandora:WADMContainerCModel). As users create annotations, their annotations and annotation containers conform to these content models. Both content models define datastreams, which are files that comprise the digital object. Every Islandora object must contain a Dublin Core (DC) and RDF datastream (RELS-EXT) datastream. Additional datastreams contain annotation-specific content.

**islandora:**WADMCModel defines two annotation-specific datastreams: WADM and WADM_SEARCH. An interface view of all the datastreams that make up a Web Annotation is depicted in Figure 2.

The WADM datastream, which is in JSON-LD format, defines the annotation body and target. We hope that storing JSON-LD will ease migration to the Fedora 4 linked-data platform.
Figure 2 - data object datastreams in an Islandora (Drupal) interface. The WADM and WADM_SEARCH datastreams are specific to the Web Annotation Utility module.

```json
{
    "@context": [http://www.w3.org/ns/anno.jsonld],
    "@id": "http://localhost:8000/islandora/object/annotation:20",
    "@type": "Annotation",
    "target": {
        "source": "http://localhost:8000/islandora/object/islandora%3A615/datastream/MEDIUM_SIZE/view",
        "format": "image",
        "selector": {
            "shapes": [
                {
                    "type": "rect",
                    "geometry": {
                        "x": "0.29",
                        "y": "0.08791208791208792",
                        "width": "0.12",
                        "height": "0.07967032967032966"
                    }
                }
            ]
        }
    },
    "body": {
        "type": "TextualBody",
        "bodytext": "An annotation, created by me!",
        "format": "text/plain"
    },
    "creator": "testannotator",
    "created: "2017-05-25T18:55:58Z"
}
```

Figure 3. JSON-LD content of the WADM datastream
The second datastream, WADM_Search, contains an XML representation of the JSON-LD data in the WADM datastream for indexing purposes. By default, Islandora’s use of Solr leverages an application called Gsearch, which only indexes XML and text content. By creating an XML representation, we ensure that data (like the annotation author’s username and the content of the annotation) is properly indexed. This approach also permits users familiar with Islandora’s Gsearch/Solr framework to make modifications to the indexing of annotations. By indexing annotations in Solr, users and site builders can index, retrieve and display annotations for either a specific object or repository-wide. Users can also use extensions such as Islandora Solr Views Module (which permits indexed Solr fields to be displayed/visualized or downloaded using Drupal’s Views Framework) [17][18].

![Image 74x135 to 542x253](Image 74x395 to 343x568)

**Figure 4.** Sample content of the WADM_SEARCH datastream

The *Web Annotation Utility Module* also makes modifications to Fedora 3’s default RDF datastream (RELS-EXT) to add RDF that associates the annotation to an annotation container object (islandora:WADMContainerCModel). Figure 5 shows the content of the RDF datastream for an annotation object, which is connected to an annotation container using `fedora:memberOfAnnotationContainer`.

```
<annotation>
  <title>Annotation for islandora:4</title>
  <target>islandora:4</target>
  <creator>testauthenticateduser2</creator>
  <textvalue>Child created by user2</textvalue>
</annotation>
```

**Figure 5.** Example of Fedora’s default RDF datastream for an annotation object. The `fedora:memberOfAnnotationContainer` statement contains the reference to the Annotation container.

```
<rdf:RDF>
  <rdf:Description rdf:about="info:fedora/annotation:20">
    <fedora-model:hasModel rdf:resource="info:fedora/islandora:WADMCMModel"/>
    <fedora:memberOfCollection rdf:resource="info:fedora/islandora:WADMCMModel"/>
    <fedora:memberOfAnnotationContainer rdf:resource="info:fedora/annotation:18"/>
  </rdf:Description>
</rdf:RDF>
```
**islandora:**WADMContainerCModel defines one annotation-specific datastream, WADMContainer. An interface view of all the datastreams that make up a Web Annotation Container is depicted in Figure 6.

![Figure 6. WADMContainer datastream specific to the Web Annotation Utility module.](image)

This datastream, WADMContainer, stores the JSON-LD representation of the container.

```
{
    "@context": [
        "http://www.w3.org/ns/anno.jsonld",
        "http://www.w3.org/ns/ldp.jsonld"
    ],
    "@id": "http://localhost:8000/islandora/object/annotation:19",
    "@type": ["BasicContainer", "AnnotationCollection"],
    "total": "1",
    "label": "annotationContainer for islandora:7",
    "first": {
        "id": "http://localhost:8000/islandora/object/islandora%3A7/annotations/?page=0",
        "type": "AnnotationPage",
        "items": [
            "annotation:26",
            "annotation:27"
        ]
    }
}
```

**Figure 7. JSON-LD content of the WADMContainer datastream**

The *Web Annotation Utility Module* also makes modifications to Fedora 3's default RDF datastream (RELS-EXT) to add RDF that associates the container to the object being
annotated. Figure 8 shows the RDF content of this datastream in the Container Object, which is connected to the object being annotated using `fedora:isAnnotationContainerOf`.

```
<rdf:RDF>
  <rdf:Description rdf:about="info:fedora/annotation:25">
    <fedora-model:hasModel rdf:resource="info:fedora/islandora:WADMContainerCModel"/>
    <fedora-isMemberOfCollection rdf:resource="info:fedora/islandora:WADMContainerCModel"/>
    <fedora-isAnnotationContainerOf rdf:resource="info:fedora/islandora:7"/>
  </rdf:Description>
</rdf:RDF>
```

**Figure 8.** Example of Fedora’s default RDF datastream for annotation container. The `fedora:isAnnotationContainerOf` statement contains the reference to the object being annotated.

As this section describes, RDF statements in the container object connect it to the object being annotated. Similarly, RDF statements in the annotation connect it to the annotation container. For each object that is annotated, there will be one annotation container and potentially multiple annotations.

**Annotation Creation Workflow**

The diagram in Figure 9 provides a high-level overview of the entire annotation creation process. When a user views and creates an annotation, one of two things occur:

1. An annotation is created and linked back to an existing annotation container, or

2. An annotation container is created concurrently with the annotation (when the annotation is the first one made on an object).
Not only does this structure reflect the specifications of the Web Annotation Data Model, it also facilitates better performance when bringing back an annotated object's related annotations.

![Diagram of annotation workflow]

**Figure 9.** Annotation creation workflow. An annotation container and annotation are created and related to one another. Subsequent annotations will be related to the annotation container for the object.

**Drupal/Islandora Viewers and Annotation Creation**

In building the Islandora Web Annotation Utility, we tried to reuse as much existing community work as possible. On the front end, all annotation authoring is managed by Drupal. Drupal is a popular content management system for creating complex sites. The wide adoption and development of Drupal is an impetus for institutions installing Islandora. Drupal has also inspired Islandora developers, who sometimes extend existing Drupal modules or adopt Drupal module approaches, to provide enhanced features for Islandora. One example of this type of an approach is the Islandora Context Module[19]. In Drupal, the Context module[20] allows for site builders to author particular contexts in which rules are executed. Each time a page is loaded, the Context module seeks relevant conditions and executes rules that have been defined by an administrator. Mark Jordan, Head of Library Systems at the W.A.C. Bennett Library at Simon Fraser University, built the Islandora Context Module to extend the Context module and enables rules to be executed on Islandora (Fedora) specific contexts, such as a keyword in an object datastream, or when the object is of a particular content model.

The Islandora Web Annotation Utility uses the Islandora Context module to load javascript and CSS files and target specific Islandora viewers and content models, enabling annotation. Work thus far has focused on targeting simple images, large images, book pages, video, audio and oral history content.
Figure 10 outlines the setup of the module in a Drupal environment. Site builders or administrators define content-specific contexts, depending on what type of content they wish to make it possible to annotate. Users are then given annotating “roles” in Drupal to view, create, edit or delete their own annotations and/or the annotations of others. Users can then begin annotating repository content.

Figure 10. A site administrator defines contexts for specific content types and assigns appropriate permissions to annotating users

The Islandora Context Module provided us a way of targeting specific types of repository content. Existing annotation libraries could then be leveraged to provide annotation tools depending on the type of content being annotated. Local use cases guided our decision to prioritize the ability to annotate objects using the OpenSeadragon viewer (which is a zoomable viewer for large images) and the Video.js player, as well as our Oral History object player (which is based on, and utilizes, our video.js player).

OpenSeadragon and Annotorious
OpenSeadragon is a open-source javascript viewer for web content that supports high resolution zoomable images [21]. In Islandora, the OpenSeadragon module [22] enables the viewer to work with an Islandora installation by providing a Djatoka [23] tile source for the viewer.

Administrators can configure the OpenSeadragon viewer to display Islandora-managed large image and page content. The Islandora Web Annotation Utility module uses the Annotorious javascript library, developed “under the leadership of the Austrian Institute of Technology,” [24] to provide users with the ability to annotate objects using the OpenSeadragon viewer. When a user encounters an image object and has been granted permissions to annotate this content model, two new buttons appear alongside OpenSeadragon’s default navigation buttons (see Figure 11), allowing the user to view and hide annotations and to create new annotations.

![OpenSeadragon Interface](image)

**Figure 11** user experience when hovering over an annotation created in the OpenSeadragon viewer. Each annotation is identified by a number that corresponds to a legend below the image.

When a new annotation is created, it is assigned a number in the interface which corresponds to the annotation text that is displayed beneath the metadata (Figure 12). These numbers are a UI guide only and assigned when the user visits the page. Hovering over the image reveals the zoom features of the OpenSeadragon viewer, as well as the text of annotations. Rich text for
image annotations is not currently supported, though this is supported in the Web Annotation specification and is on our Roadmap.

![Image](image.png)

**Figure 12.** Annotations and their corresponding number in the interface.

While the annotation module provides a listing of annotations beneath the viewer as a default, Islandora Solr Views can be configured with Contextual Filters to provide custom views of indexed annotation content, including rich text elements such as links to users and the management page for the annotation object. An example is described in the documentation for the project. When installing the module, a default drupal view that users can modify is also installed and available for use [25]. The Annotorious library also provides the annotation features for the simple image viewer in Islandora.

**Video.js and Open Video Annotations**

Video.js[26] is a viewer for Islandora that provides the default viewer for video and audio objects, as well as the core of the Islandora Oral Histories Solution Pack [15]. The front end uses the OpenVideoAnnotation javascript library developed by Center for Hellenic Studies to enable users to annotate objects in the Video.js viewer. OpenVideoAnnotation makes use of Annotator.js, which is popular library that underpins the Hypothes.is project [27] [28] [29] [30].

When a user has permissions to annotate media assets, new buttons provided by the plugin appear in the video.js toolbar (Figure 13). Users select portions of the timeline to annotate. Rich text is supported for this module.
**Figure 13** Video.js viewer, configured for annotation with active green annotation selected by user. Three new buttons available. The button highlighted in this screenshot shows the regions of the video that have been annotated.

In addition to creating annotations, users can also see where annotations have been added to the video via yellow bars across the timeline of the media being annotated. Users can also view statistics for which portions of the media timeline have been most heavily annotated. As with the OpenSeadragon viewer, annotations are viewable in a block. However, unlike the automatic block provided with the Simple and Large Image viewer, the view is powered by Solr indexing of annotation content in Fedora (Figure 14).
In both players, permissions provided by the library are respected by the Islandora ecosystem and created objects conform to the same object model.

**Idiosyncrasies in the Islandora Ecosystem**

Our team is committed to robust testing and documentation for the module, as well as more advanced administration features and rational integration with other Islandora-contributed modules. We acknowledge that the module has weaknesses in this regard. Strictly speaking, the module bypasses some core Islandora permissions, in that only the core “View Objects” permission is required to utilize the Web Annotation specific permissions. This means that a user can create annotations, which are Islandora objects, without having the core Islandora “Create Objects” permission. However, we have not encountered any practical setbacks as a result of this situation.

The biggest practical weakness we have discovered is that the module does not work as expected with Islandora’s *Simple Workflow Module*[31]. This is because that module requires that objects be created in Fedora’s “inactive” state, and our annotation containers need to be “active” objects for the module to work. However, we need to support annotations moving through a workflow and tackling this issue is also of concern to our group for future versions of the software.
Current Implementation and Release Schedule

Early local uptake of the module is promising. A UTSC English Professor (Anne Milne) is using the module in an upper-level undergraduate classroom, where students annotate the detailed engraved plates created by early eighteenth-century artist William Hogarth, as a method for both exploring and describing his historical and cultural context. Since winter 2015, senior students have curated around 200 annotations on photographs of four “series” of engraved plates on the Hogarth website [32]. A local oral history project led by history professor Chris Berkowitz also aims to use the module to enrich existing histories. Chris Berkowitz and her colleagues have played a seminal user role in the development of these features. Oral Histories have been collected by Chris Berkowitz’s class since 2013, representing a large dataset ripe for enrichment in future years.

The current release of the module is available on Github [3] or our main website [33]. The currently released version has been tested against the Islandora 7.x-1.9 release. Details of our testing procedures, including supported browsers and themes, are available in the module repository. We aim to test features after the release of each Islandora version (currently twice yearly, spring and fall). We are open to hearing from other potential implementers or collaborators regarding this work.

Conclusion

As Web Annotation continues to become a refined concept associated with community best practices, it may be possible to further consider how to integrate annotation functions into existing repository and digital library frameworks by leveraging tools and data models that are already in place. While our focus is Islandora and Fedora-focused, we expect to see similar work in other systems. We also believe that all practical work to implement the specification will contribute to additional uptake and use of the standard across systems. There is great potential for digital library programs to enrich existing collections and facilitate web-based scholarly research and collaboration. For Islandora users, we hope this may become an approach to annotation that can be agreed upon and supported by the wider Islandora community.

Acknowledgements

We would like to acknowledge Mark Jordan, Head of Library Systems W.A.C. Bennett Library at Simon Fraser University for developing the Islandora Context Module. We would also like to acknowledge Lingling Jiang, Metadata & Systems Librarian at Sheridan College, for her initial work on the Web Annotation content models (initially part of the Oral History Solution Pack).
References


