

Article

Towards Annotopia—Enabling the Semantic Interoperability of Web-Based Annotations

Jane Hunter * and Anna Gerber

School of Information Technology and Electrical Engineering, The University of Queensland, St. Lucia, QLD 4072, Australia; E-Mail: a.gerber@uq.edu.au

* Author to whom correspondence should be addressed; E-Mail: j.hunter@uq.edu.au; Tel.: +61-7-3365-1092.

Received: 21 June 2012; in revised form: 7 August 2012 / Accepted: 24 August 2012 /

Published: 30 August 2012

Abstract: This paper describes the results of a collaborative effort that has reconciled the Open Annotation Collaboration (OAC) ontology and the Annotation Ontology (AO) to produce a merged data model [the Open Annotation (OA) data model] to describe Web-based annotations—and hence facilitate the discovery, sharing and re-use of such annotations. Using a number of case studies that include digital scholarly editing, 3D museum artifacts and sensor data streams, we evaluate the OA model’s capabilities. We also describe our implementation of an online annotation server that supports the storage, search and retrieval of OA-compliant annotations across multiple applications and disciplines. Finally we discuss outstanding problem issues associated with the OA ontology, and the impact that certain design decisions have had on the efficient storage, indexing, search and retrieval of complex structured annotations.

Keywords: annotation; interoperability; semantic web; ontology

1. Introduction and Background

Annotating documents is a core and pervasive practice for scholars across both the humanities and sciences. It can be used to organize and share existing knowledge or to facilitate the creation of new knowledge. Annotations may be used by an individual scholar for note-taking, for classifying data and documents, or by groups of scholars to enable shared-editing, collaborative analysis and pedagogy.

Although numerous systems exist for annotating digital resources [1], many of these systems can only be used to annotate specific collections, are based on proprietary annotation models, and were not designed to support open access to or sharing of annotations. Scholars working across multiple content repositories often have to learn to use a number of annotation clients, all with different capabilities and limitations, and are unable to integrate their own annotations across collections, or to reference or respond to annotations created by colleagues using different systems. This presents a barrier for collaborative digital scholarship across disciplinary, institutional and national boundaries.

Given the importance of annotation as a scholarly practice, these issues, combined with the lack of robust interoperable annotation tools, and the difficulty of migrating annotations between systems, are hindering the exploitation of digital resources by scholars across many disciplines.

1.1. The Open Annotation Collaboration (OAC) Data Model

The OAC project is a collaboration between the University of Illinois, the University of Queensland, Los Alamos National Laboratory Research Library, the George Mason University and the University of Maryland, which is funded by the Andrew W. Mellon Foundation. The OAC project was established to define a framework to enable sharing and interoperability of scholarly annotations on digital resources.

The collaboration has produced an RDF-based Open Annotation data model [2], which draws on a number of previous efforts including: the W3C's Annotea model [3], Agosti's formal model [4]; SANE Scholarly Annotation Exchange [5]; and OATS (The Open Annotation and Tagging System [6]. An analysis of these existing models reveals that on the whole, they have not been designed to be both Web-centric and resource-centric. Moreover, they have modeling shortcomings that include: the preclusion of existing Web resources (regardless of media type) from being the content or target of an annotation; the inability to publish annotations as independent, stand-alone Web resources; the lack of support for attaching a single annotation (body) to multiple targets.

1.2. The Annotation Ontology (AO)

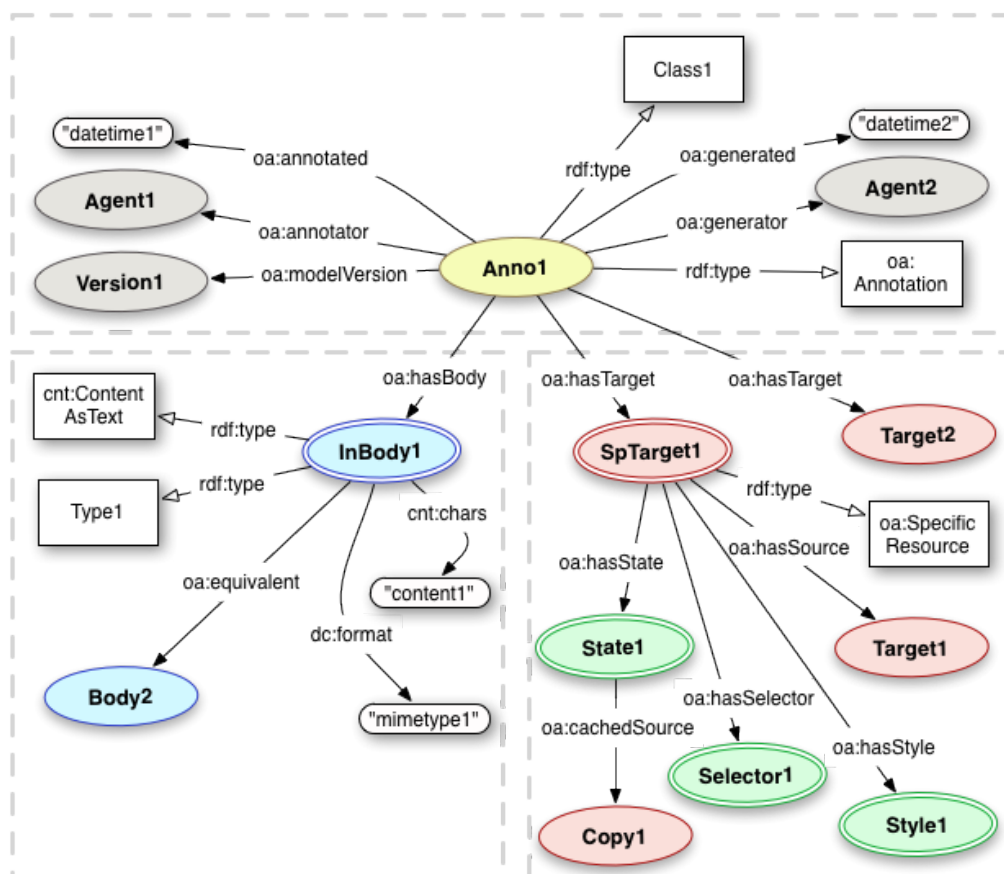
Developed concurrently, but independently to the OAC model, the Annotation Ontology [7] provides a model for annotating scientific documents on the web. The AO model draws on the Annotea model and reuses the Provenance, Authoring and Versioning (PAV) ontology for annotation provenance. AO was designed to be integrated with the Semantic Web Applications in Neuromedicine (SWAN) ontology used for hypothesis-based representation of scientific discourse. It was also designed to integrate easily with existing scientific ontologies and vocabularies such as those expressed using Simple Knowledge Organization System (SKOS) and social web ontologies such as Semantically-Interlinked Online Communities (SIOC). AO was originally designed to enable biomedical domain ontologies to be used to annotate scientific literature but its creators believe it is more widely applicable to Web resources in general.

1.3. The Merged Open Annotation Model (OA)

The W3C Open Annotation Community Group was established in 2011 to specify an extensible, interoperable framework for representing annotations, by aligning the OAC and AO data models. Figure 1 illustrates the draft Open Annotation (OA) model [8], which has emerged from the alignment of the two models. A comparison of the key differences between the OAC and OA models is presented in Table 1.

In Figure 1, an OA *Annotation* instance, *Anno1* represents the reification of an *annotates* relationship between a *Body* (the comment or data being attached by the annotator) and one or more *Target* resources [where the Body is “about” the Target(s)]. Body and Target resources can be of any type e.g., image, audio, video or text-based formats including HTML. Annotation provenance is recorded via properties, (see top section of Figure 1), that describe the context in which the annotation was created, including the time of creation and the human or software agent responsible for creating the annotation. Annotation bodies can be identified by URI or included as inline content (*i.e.*, where the content is embedded in the *Annotation*) using properties from the W3C’s “Representing Content in RDF” [9] ontology, as illustrated in the bottom left of Figure 1.

Figure 1. The Open Annotation (OA) model.



Note: Figure 1 republished from the OA Core Data Model, under the terms of the W3C Community Contributor License Agreement.

Table 1. Comparison of Open Annotation Collaboration (OAC) and OA models.

Feature	Model	
	OAC	OA
Provenance	Recommends use of Dublin Core <i>creator, created etc.</i>	Defines provenance properties, distinguishing between annotator and generator
Annotation types	Defines two subtypes of <i>oac:Annotation</i> : (1) <i>oac:Reply</i> and (2) <i>oac:DataAnnotation</i> . OAC encourages subtyping.	Common subtypes described in OAX rather than the core model. Provides guidelines for subtyping. OA recommends explicitly including <i>oa:Annotation</i> type for all annotations.
Equivalent serializations	No equivalent concept in OAC.	Defines <i>equivalent</i> property for maintaining a relationship between re-published copies of an annotation
Body	Does not specify cardinality of annotation bodies.	Explicitly allows annotations without a body, does not allow multiple bodies.
Resource segments	<ul style="list-style-type: none"> Recommends use of Fragment URIs for target and body resources, with a <i>dcterms:isPartOf</i> relationship to base resource. <i>ConstrainedTarget</i> and <i>ConstrainedBody</i> with <i>oac:constrains</i> allow other types of segment descriptions to be specified via the <i>oac:constrainedBy</i> property. Defines a number of constraint types including <i>oac:OffsetRangeConstraint</i>, <i>oac:PrefixPostfixConstraint</i>, <i>oac:SvgConstraint</i>, <i>oac:WebTimeConstraint</i> and <i>oac:ContextConstraint</i>. 	<ul style="list-style-type: none"> <i>oa:SpecificResource</i> replaces <i>oac:ConstrainedTarget</i> and <i>oac:ConstrainedBody</i> <i>oa:hasSource</i> property replaces <i>oac:constrains</i>. <i>oac:Constraints</i> are replaced with <i>oa:Selectors</i>. Fragment URIs are disallowed for <i>oa:hasBody</i> and <i>oa:hasTarget</i>. <i>oa:FragmentSelector</i> is defined for representing fragments instead. All other types of selectors are described in OAX rather than the core model. Allows multiple alternative selectors.
State	No equivalent concept in OAC.	Defines <i>oa:hasState</i> which associates an <i>oa:State</i> with an <i>oa:SpecificResource</i> to allow clients to retrieve the correct representation of the resource (e.g., version, format, language etc).
Style	No equivalent concept in OAC.	Defines <i>oa:hasStyle</i> which associates an <i>oa:Style</i> with an <i>oa:SpecificResource</i> to enable hints to be provided to the client recommending how to display the <i>Annotation</i> and/or <i>Selector</i> .
Extensions	No equivalent concept in OAC.	Splits the model into a stable core (OA), and extension (OAX), which may change. OAX includes subtypes of <i>Annotation</i> , <i>Selector</i> , <i>State</i> and <i>Style</i> , <i>hasSemanticTag</i> and support for Named Graphs for structured data.

Many Annotations apply to segments of resources, such as a region of an image, or a paragraph of text within a larger document. In these cases, instead of targeting the URI that identifies the entire resource, an Annotation will target a *SpecificResource* representing the segment of the resource. The extent of the segment is described by a *Selector* and the complete resource is specified by *hasSource*. The OA community group is working on defining a set of common subclasses of *Selector*, including *FragmentSelector*, which can be used to specify URI fragments using existing fragment schemes

including HTML/XML IDs, XPointers, or W3C Media Fragments, and *SVGSelector* for describing image regions. *State* captures contextual information about a Target resource when the annotation was attached, to assist clients in retrieving the correct version or representation of the Target resource as it existed at the time of annotation. For example, the *State* information ensures the appropriate MIME type is retrieved, if the same URI can return different representations via HTTP content negotiation. *Style* is used to provide hints to an annotation client about how an annotation should be displayed, which is useful in applications where color or formatting of the annotation markers needs to be persistent. *Selectors*, *States* and *Styles* can be specified inline (as for the body of this annotation), or by reference using a URI or URN. *Selectors*, *States* and *Styles* can also be applied to a *SpecificResource* representing the *Body* of an *Annotation*.

In addition to the core classes and properties which are defined in the Core Open Annotation Namespace (“oa:”) [10], the W3C Annotation Community Group are also working on an Extension ontology (namespace “oax:”) [11] that defines additional sub-classes and properties that are specific to certain common use cases and content media types. These extensions include: oa:Annotation subClasses (e.g., *oax:Bookmark*, *oax:Comment*, *oax:Description*, *oax:Highlight*, *oax:Question*, *oax:Reply*, *oax:Tag*); annotation types (*dctypes:Dataset*; *dctypes:Image*; *dctypes:MovingImage*; *dctypes:Sound*; *dctypes:Text*); media-specific Selectors (*oax:TextOffsetSelector*; *oax:TextQuoteSelector*; *oax:SvgSelector*); oa:Style subClasses (*oax:CssVlaueStyle*, *oax:XsltStyle*) and additional properties (e.g., *hasSemanticTag*—to be used for tags that have URIs). It is anticipated that the Core data model will be mostly static whilst the Extensions specification will provide the flexibility to accommodate changes and refinements based on community feedback.

In the remainder of this paper, we apply the Open Annotation (OA) model to a number of scholarly annotation use cases spanning several disciplines, in order to evaluate its capabilities. We also provide a discussion of some of the problematic and open issues associated with the OA model.

2. Case Studies

2.1. Annotations for Electronic Scholarly Editions

Scholarly editions are the outcome of detailed study of a specific literary work or collection of shorter works such as poems or short stories. When preparing a scholarly edition, scholarly editors aim to provide a comprehensive description of the history of the literary work(s) including information about significant versions and physical forms. The IFLA FRBR model, which allows bibliographic entities to be described as Works, Expressions, Manifestations or Items [12] can be used as a foundation for describing these versions. Versions can be represented as FRBR Expressions, while each issue of a given version can be represented as a FRBR Manifestation. Physical characteristics of specific physical objects such as manuscripts are described at the FRBR Item level. In addition, a scholarly edition usually comprises a textual essay, textual notes that analyse variations and editorial decisions, and a textual apparatus that is compiled to record the alterations made between different versions of the work. Annotations can be used to document these textual notes and variations, and can provide an additional layer of information about the documents being studied, and the people or organizations who were involved in the production of the work over time. Annotations in the form

of explanatory notes may also address the content of the text, identifying such things as allusions to other works, historical contexts and stylistic significance. Modern scholarly editions are increasingly collaborative exercises that involve a team of editors, advisors and editorial board members dispersed globally.

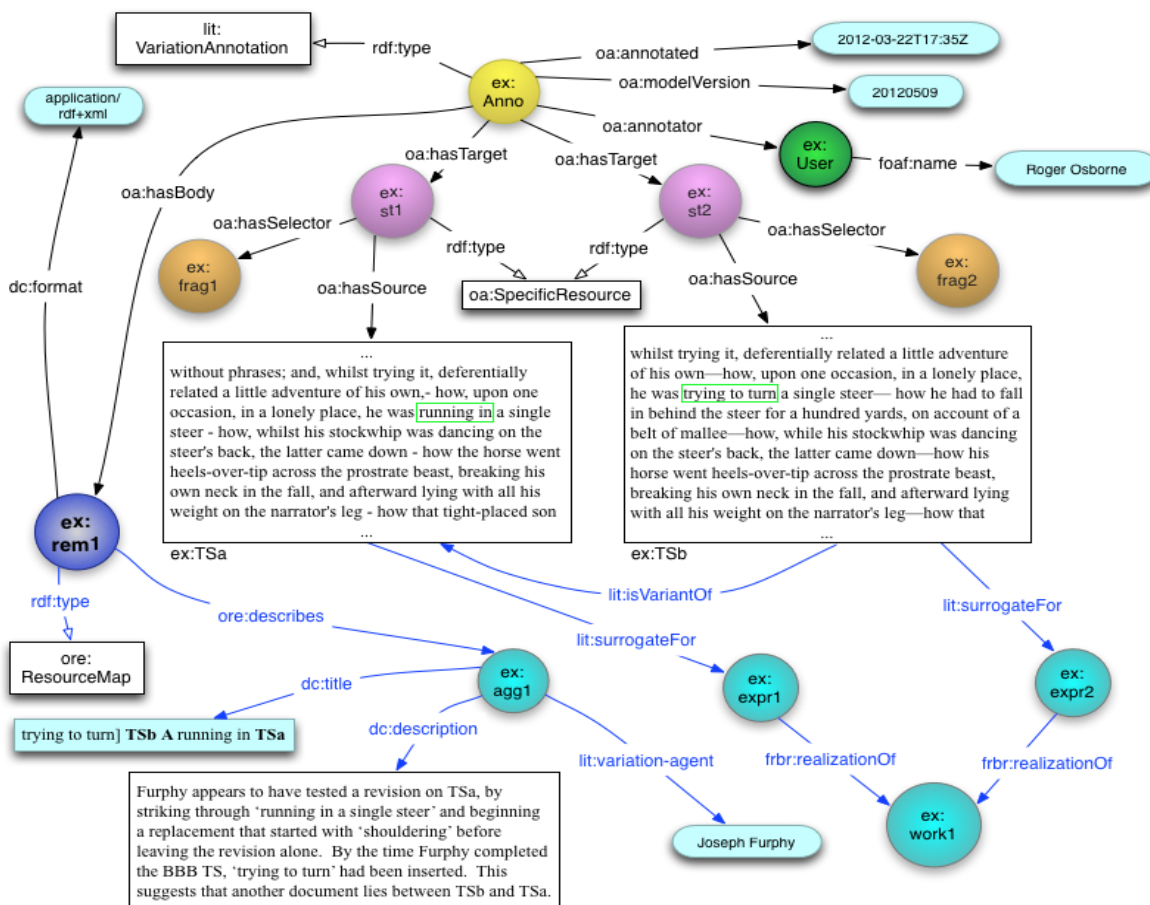
The AustESE project [13] is a collaboration between The University of Queensland, University of NSW, Curtin University, University of Sydney, Queensland University of Technology, Loyola University, Chicago and the University of Saskatchewan, which aims to develop a set of interoperable services to support the production of electronic scholarly editions by distributed collaborators in a Web 2.0 environment. The annotation service that we are developing for the AustESE project enables scholarly editors to:

- Create annotations that relate transcripts with facsimiles;
- Attach notes to text and image selections;
- Reference secondary sources;
- Annotate textual variations—record information about the reason for or the source of the variation;
- Engage in collaborative discussion about texts through comments, questions and replies.

To support these use cases, we have defined several custom subclasses of the OA *Annotation* class including: *ExplanatoryNote* to provide explanatory commentary on selected characters, words, paragraphs, sections etc.; *TextualNote*, to document or provide support for editorial decisions; and *VariationAnnotation* to describe textual variations between multiple versions of a literary work. When a scholarly editor creates a *VariationAnnotation*, the content of the Annotation includes an assertion that there is a directional relationship between two versions (the original and the edited); and commentary and/or metadata (e.g., the responsible agent, the date of the variation, the reason for the variation), which applies to the asserted relationship: it does not apply to the versions individually.

Figure 2 shows an example of a *VariationAnnotation*. The Body (ex:rem1) is an OAI ORE Resource Map [14] that reifies the *isVariantOf* relationship that the scholar is attaching between the original and variant texts. Our use of ORE Resource Map to reify a relationship between resources was inspired by the way that the OA Annotation class reifies the *oa:annotates* relationship between body and target resources. The properties and relationships displayed in blue (at the bottom of Figure 2) are encapsulated within the Body, whilst those at the top of Figure 2, exist within the *Annotation* graph. As an ORE Resource Map, the Body resource can be identified by a URI and can be reused outside of the context of the Annotation. By using ORE to represent complex scholarly annotation bodies, we hope to leverage existing tools such as LORE [15], a graphical authoring, publishing and visualization tool that we developed for authoring ORE Resource Maps.

Figure 2. oa: Annotation describing Textual Variation.



In Figure 2, both of the target resources are transcripts rendered as HTML, however, scholars may also annotate other digital surrogates of the same version of the work (including transcripts) in alternate formats such as plain text, TEI XML or scanned page images. Each target transcript is associated with a FRBR Manifestation via custom ontology properties, which identify the bibliographic entities for the versions of the work under study. These properties can be traversed by SPARQL queries to retrieve annotations across all digital surrogates of the same FRBR Manifestation.

A URI identifying the person who created the annotation, as well as timestamps for when the annotation was created or modified are recorded as properties of the Annotation (see Figure 3). Also associated with each creator URI is their FOAF name property, which is used for display purposes. The OA model allows metadata such as *dcterms:creator* and *timestamps* (associated with *dcterms:created* and *oa:annotated*) to be specified for each Body and Target resource as well as the Annotation object. This means that the author of an annotation Body can be different to the author of the Annotation object, allowing scholars to create Annotations in which the Body comprises content created by other scholars. An example of this type of reuse is the creation of Annotations that use selected notes from an existing electronic scholarly edition as commentary.

Figure 3 shows an OA model instance associating a textual note originally created by Paul Eggert for a digital Scholarly Edition of Ned Kelly’s Jerilderie Letter [16]. By attaching *dcterms:creator* and *dcterms:created* to the body resource, the original scholar who created the note and the date it was created can be clearly specified. The person who has created the new Annotation (Damien Ayers) is

attributed, since they discovered and reused the original content, and may have attached tags or other metadata to the Annotation to aid discovery or classification.

Figure 4 above illustrates how collaborators can generate a discussion around a particular piece of text by replying to existing annotations. This example only shows a single reply, however it is possible to have a chain of replies in which each previous *Reply* is the *Target* of the next *Reply*.

Figure 3. OA instance for textual note.

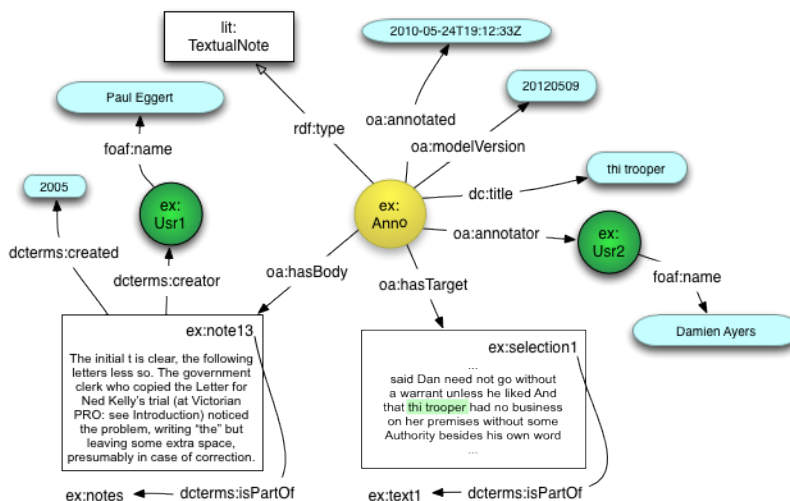
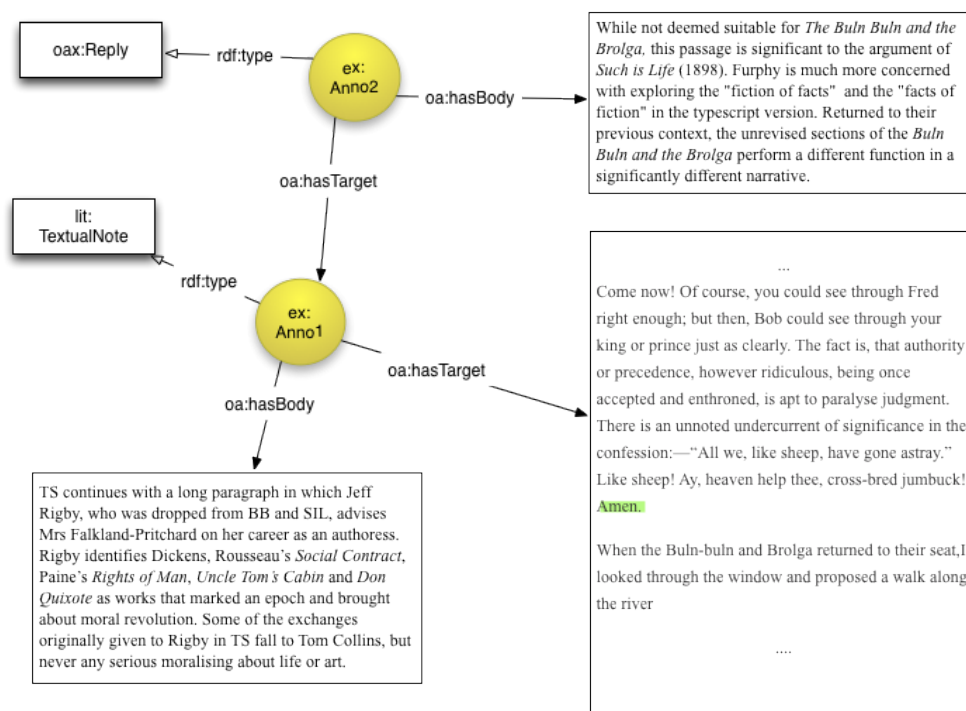


Figure 4. OA instance illustrating a reply to an existing Annotation/TextualNote.



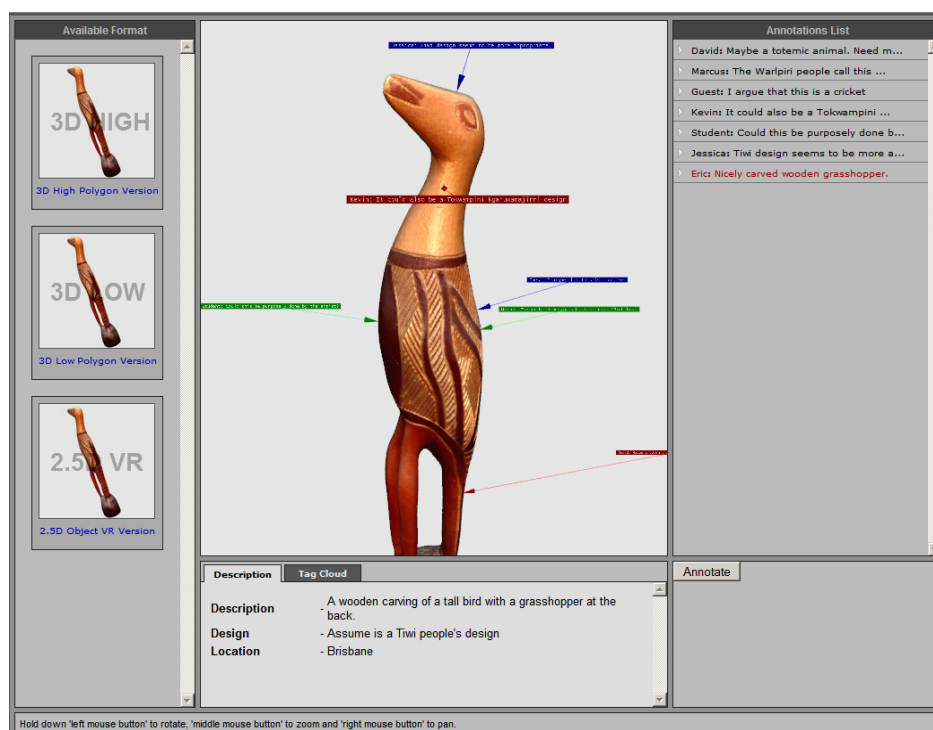
2.2. Migrating Annotations across Multiple Representations of Museum Artefacts

The 3D Semantic Annotations (3DSA) project [17] aims to develop simple, semantic annotation services for 3D digital objects (and their sub-parts) that will facilitate the discovery, capture,

inferencing and exchange of cultural heritage knowledge. The 3DSA annotation tool enables users (curators, scholars, students and the general public) to search, browse, retrieve and *annotate* 3D digital representations of cultural heritage objects via a Web browser. The first step involves generating a collection of 3D models of artefacts from the UQ Antiquities Museum (Greek Vases and Roman Sculptures) and the UQ Anthropology Museum (Indigenous carvings from the Wik community in Western Cape York). The 3D representations have been created using a Konica Minolta 9i 3D laser scanner.

The 3DSA annotation client, shown in Figure 5, is a Web application that allows ontology-based (semantic) tags and/or free text comments to be attached to 3D objects. Annotations can be attached to the whole object or to a point, surface region or 3D volumetric segment of the object. The central panel provides an interactive 3D viewer that allows the user to pan, zoom and rotate the 3D object. Existing annotations on the object are indicated by the colored markers attached to the object, and are listed on the right-hand side. Clicking on an annotation on the RHS, rotates and zooms the object, so that that selected annotation is central and its details described in the panel at the bottom centre.

Figure 5. 3DSA Annotation client.



The laser scanner produces archival quality models, with each model being on average between 50–200 MB, containing 0.5–2 million polygons and with surface textures scanned as high resolution bit maps. However, many of the users accessing the collection from remote locations have limited bandwidth, poor graphics capabilities and limited compute power—and hence are unable to render very high resolution, archival-quality models. In order to support users with variable Internet connectivity and computing resources, each model is converted into a high resolution and low-resolution X3D format, as well as a “2.5D” FlashVR file constructed from a sequence of 2D images. Table 2 outlines the characteristics of the three formats for the web.

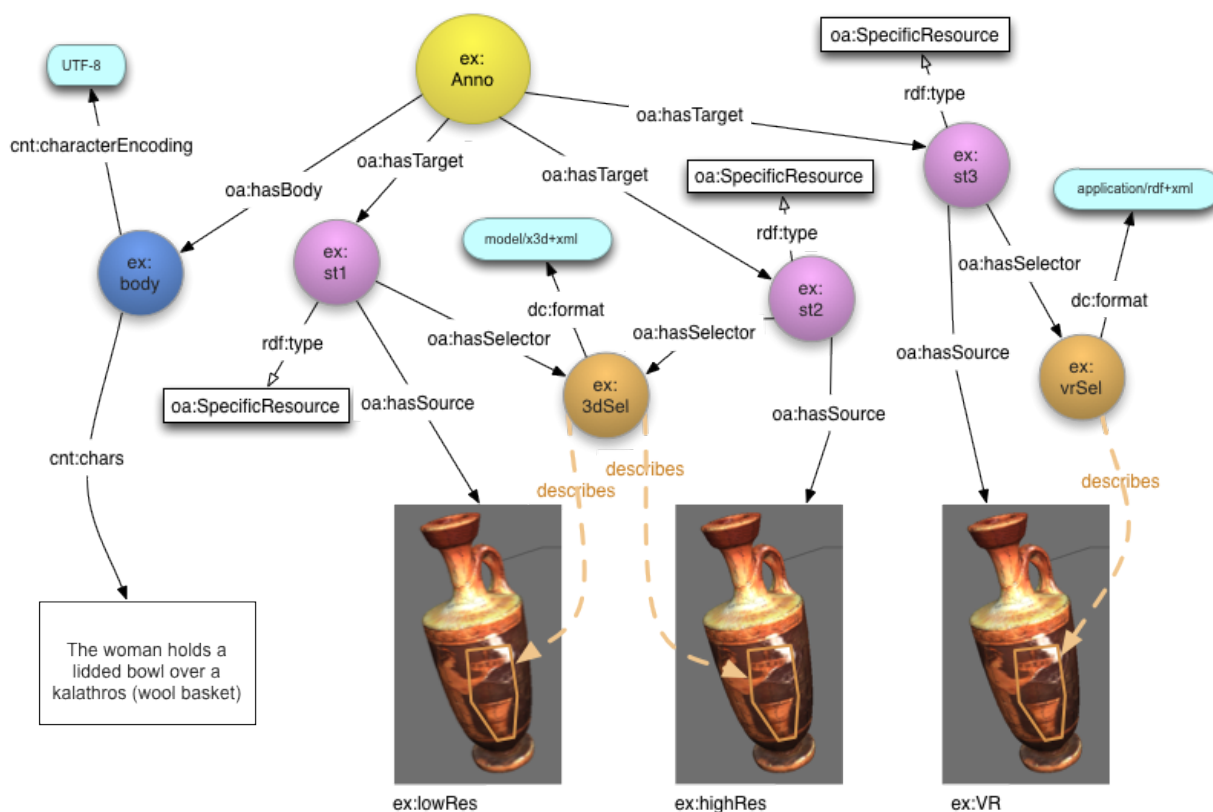
Table 2. 3D object characteristics.

Attribute	Format		
	High quality 3D	Low quality 3D	2.5D VR
Format	X3D	X3D	Flash
File size	18–21 MB	4–6 MB	1 MB
Polygon count	0.3 million	65,000	N/A

In order to allow users to collaborate regardless of which format they are using to access the museum objects, 3DSA supports the automatic migration of annotations between the high res 3D model, low res 3D model and 2.5D representations. The annotation client automatically migrates annotations attached to a point or region on one representation of the model across to the other formats. In the case of the X3D format files, the segments being annotated are identical as the 3D representations share the same local matrix. However, when migrating to the FlashVR image sequence, points need to be projected from a 3D space (identified using X, Y and Z co-ordinates) onto a 2D plane (X and Y only).

Figure 6 illustrates an OA model instance for a sample annotation after it has been migrated across the three formats. Rather than having to manage three separate annotations, and ensure that their contents remain synchronized, the OA model allows all of the derivative models to be targeted within a single annotation. The body of this annotation is specified inline using *cnt:chars* (to represent the content) and *cnt:characterEncoding* (to specify the type of the content) from the “Representing Content in RDF” ontology.

Figure 6. Instance of OA model for 3DSA.



The 3DSA client also captures the co-ordinates, rotation, light position and other related data, which is used to recreate the precise view, zoom level, and position at which the annotation was originally attached. There is no standard for addressing segments within 3D models, and as there may be many different values stored to represent each segment including co-ordinates, transformation matrixes etc, these 3D *Selectors* are not well suited to being encoded in a URI fragment, in the style of the W3C Media Fragments [18] or XPointers.

Consequently, we have had to define custom subclasses of *Selector* to describe points, surface regions and segments of a 3D or 2.5VR model. In the example shown in Figure 3, the two 3D format targets share the same X3D-format Selector (*3dSel*) for a given surface region, however the 2.5VR format requires a different selector (*qtvrSel*) because the data has been transformed to apply to the 2D plane.

In addition to having to support selectors that precisely specify points, 2D surface regions or 3D volumetric segments on 3D digital models, there is also a need to attach structured annotations that comprise both ontology-based (semantic) keywords (e.g., “attic, lekythos”) as well as full-text descriptions “The woman holds a lidded bowl over a large kalathos”. The semantic tags would be attached using the “oax:hasSemanticTag” property and the textual description would be represented as an Inline Body (since it does not have a URI).

The text box below illustrates how the OA model would represent such an annotation (using the Turtle RDF format).

```

@prefix oa: <http://www.openannotation.org/ns/>
@prefix ox: <http://www.openannotation.org/ns/>
@prefix cnt: <http://www.w3.org/2011/content#>
@prefix ex: <http://www.example.org/#>
ex:Anno1 a oa:Annotation;
  oa:hasBody ex:Body1;
  oa:hasTarget ex:ST1;
  oa:hasTarget ex:ST2;
  oa:hasTarget ex:ST3.
ex:ST1 a oa:SpecificResource;
  oa:hasSelector ex:3dSel;
  oa:hasSource <http://itee.uq.edu.au/eresearch/3dsa/vase1_highres.x3d>.
ex:ST2 a oa:SpecificResource;
  oa:hasSelector ex:3dSel;
  oa:hasSource <http://itee.uq.edu.au/eresearch/3dsa/vase1_lowres.x3d>.
ex:ST3 a oa:SpecificResource;
  oa:hasSelector ex:vrSel;
  oa:hasSource <http://itee.uq.edu.au/eresearch/3dsa/vase1.qtvr>.
ex:Anno1 ox:hasSemanticTag
<http://itee.uq.edu.au/eresearch/3dsa_ontology#lekythos>.
ex:Anno1 ox:hasSemanticTag <http://itee.uq.edu.au/eresearch/3dsa_ontology#attic>.
ex:Body1 a cnt:ContentAsText;
  cnt:characterEncoding "UTF-8";
  cnt:chars "The woman holds a lidded bowl over a large kalathos">.

```

2.3. Automatic Semantic Tagging of Species Accelerometry Data and Video Streams

Within the OzTrack project [19] at the University of Queensland, we are working with ecologists from the UQ ECO-Lab, providing tools to enable them to analyse large volumes of 3D accelerometry data acquired by attaching tri-axial accelerometers to animals (e.g., crocodiles, cassowaries, wild dogs), to study their behavior. The interpretation of animal accelerometry data is an onerous task due to: the volume and complexity of the data streams; the variability of activity and behavioural patterns across animals (due to age, environment, season); the lack of visualization and analysis tools; the inability to share data and knowledge between experts; the lack of ground truth data (e.g., in many cases there is no observational video); and the inaccessibility of machine learning/automatic recognition tools.

As a result of the challenges identified above, we have been collaborating with the ECO-Lab developing the SAAR (Semantic Annotation and Activity Recognition) system, with the aim being:

- To provide a set of Web services and Graphical User Interfaces (GUIs) that enable ecologists to quickly and easily analyse and tag 3D accelerometry datasets and associated video using terms from controlled vocabularies (pre-defined ontologies) that describe activities of interest (e.g., *walking, running, foraging, climbing, swimming, standing, lying, sleeping, feeding*);
- To build accurate, re-usable, automatic activity recognition systems (using machine learning techniques (Support Vector Machines))—by training classifiers using training sets that have been manually annotated by domain experts.

Figure 7 illustrates the overall system architecture of the SAAR system. SAAR uses the OA model to represent tags created manually by experts via the tagging system as well as tags that have been automatically generated by the classification model.

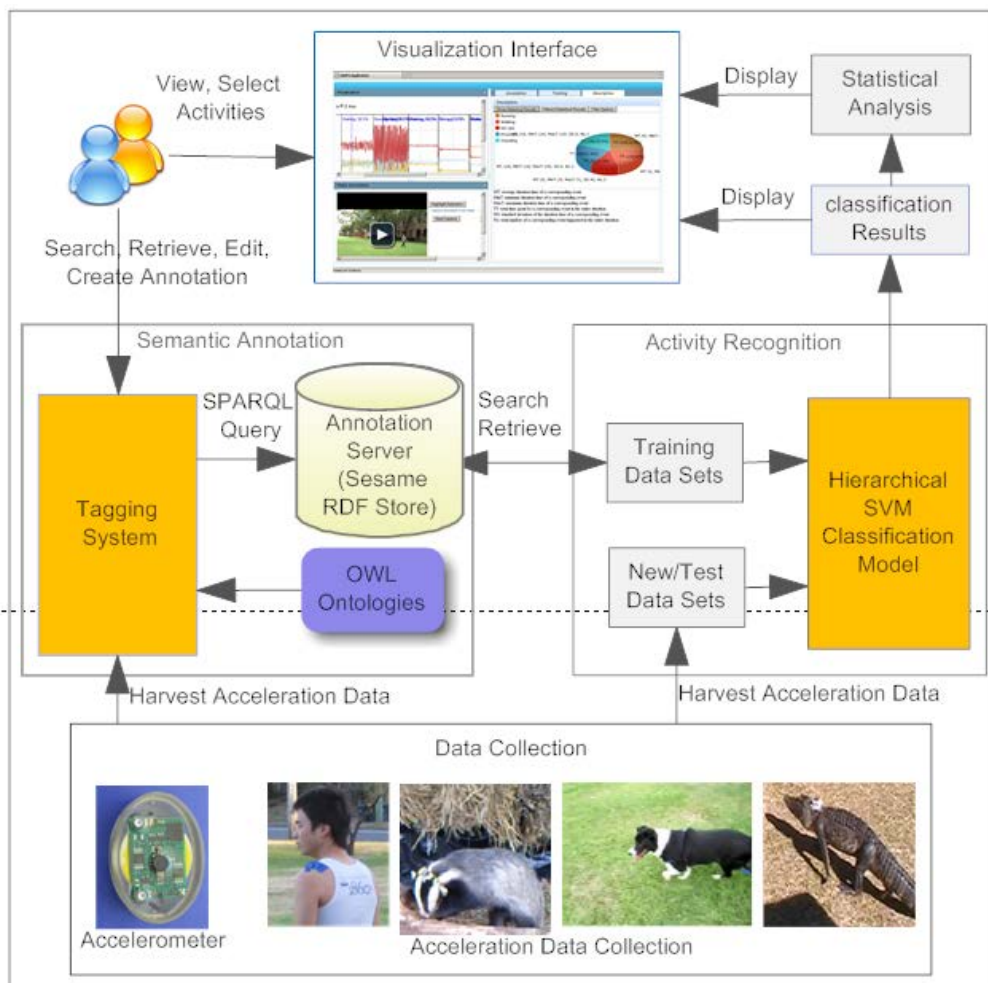
This case study is interesting from the Open Annotation data model perspective because:

- It provides a situation whereby the tags, that describe animal activities (*running, walking, swimming, climbing, standing, lying, sleeping, feeding*), are generated automatically by the SVM Classification model;
- It provides an example of the application of the OA *Fragment Selector*—tags are attached to a temporal segment of the 3D accelerometry streams as well as the associated video content where available.

The client interface comprises a Web-based Plot-Video visualization interface (see top of Figure 7) developed using a combination of AJAX, Flot (a plotting jQuery library) [20], HTML 5 Video Player library (Video.js) [21] with JavaScript. This interface enables users to interactively visualize both tri-axial accelerometer data alongside simultaneously recorded videos. Users invoke the semantic annotation service by selecting a segment of accelerometer data from the timeline or a segment of video from the video timeline, and then attaching an activity tag chosen from a pull-down menu (whose values are extracted from a pre-defined ontology). The manually created annotation is stored in a backend RDF triple store (see Section 3) implemented using the Apache Tomcat java server and Sesame 2.0. Additional annotation functions such as edit, refresh, and retrieve annotations are also supported.

The automatic activity recognition component is implemented using the Libsvm Java library. At the training stage, users can interactively search for and retrieve specific segments/annotations using the following annotation search terms: *species*, *creator*, *animal ID* and *activity_tag*.

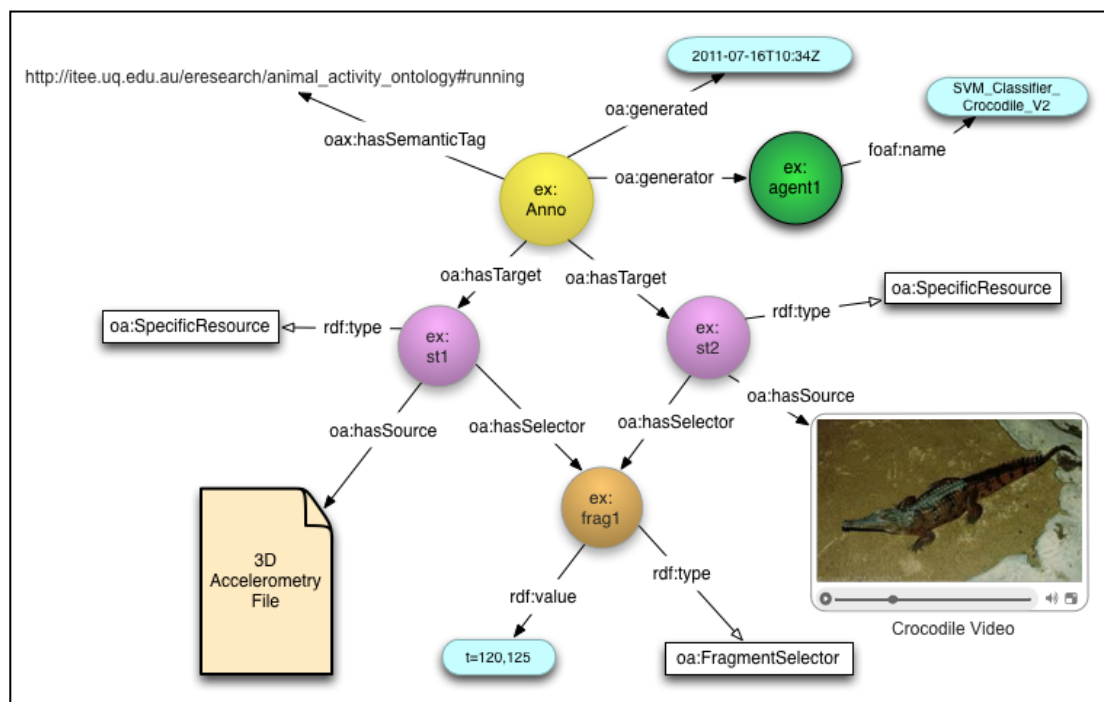
Figure 7. High level architectural view of the SAAR system.



The query is converted to the SPARQL query language which queries the annotation server. The retrieved annotations and data streams are processed to generate a set of application-dependent features that correspond to each tag. After the specific hierarchical SVM classification model is built for all of the activity tags, new tri-axial accelerometer data are input to the trained SVM classifier which then automatically tags the fresh input data (based on similarity between features). The automatically assigned tags are displayed in the timeline visualization pane, where experts can check or correct them.

Figure 8 illustrates an instance of the OA data model that represents an automatically extracted tag (“running”) that has been assigned to a specific temporal segment (from 120 s to 125 s) of accelerometry data and associated video, using an *oa:FragmentSelector* containing a W3C Media Fragment (“t = 120,125”).

Figure 8. OA representation of an automatically generated animal activity tag.



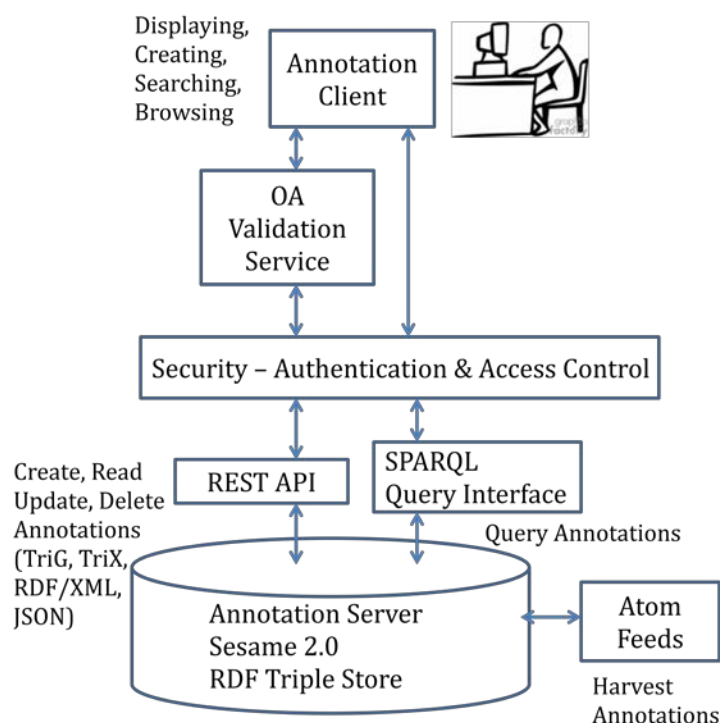
3. Implementation

Each of the case studies described above makes use of custom annotation client software, which has been developed to provide a user interface tailored to the Annotation types and user needs, specific to the community. However, by adopting the common OA model within these applications, we have been able to develop a generic Open Source Annotation repository, lorestore [22], which can be used to store, search and retrieve Annotations for all of the case studies described above.

The lorestore repository was developed as a Java Web application on top of the Sesame 2.0 RDF framework, and uses a Named Graph to store each Annotation and ORE-compliant data body. A REST API is provided to support CRUD operations on Annotations and bodies (Create, Read, Update, Delete) as well as search and retrieval by target URI or via keyword search. A SPARQL endpoint has also been implemented to enable custom queries, and ATOM feeds are implemented to allow subscription to, or harvesting of Annotations by target. The repository supports HTTP content negotiation so that Annotations can be retrieved by their identifying URI in a variety of serialization formats including TriG, TriX, RDF/XML and JSON. Authentication and user account management are implemented using the Emmet framework over Spring security, which supports basic role-based, OpenID and/or Shibboleth-based authentication. Annotations can be flagged as private, so that only the owner can view or modify their content, or locked, so that they cannot be modified. A Web-based User Interface for the repository provides a search interface and renders Annotations from their underlying RDF representation to human-readable HTML as well as diagrammatic representations. For administrators, the Web interface also supports various content management and user management operations. Figure 9 illustrates the technical components of our generic Annotation repository, lorestore.

In the near future we will begin implementation of a Web service that validates an annotation by checking its conformance with the Open Annotation Core ontology. This service will return a positive or negative message, with regard to compliance, prior to the annotation being saved in the backend repository. The validation Web service will be designed so that it can interact with any authoring tool or backend repository. It is shown in Figure 9 between the Client and the Security Layer.

Figure 9. Technical components of the lorestore annotation repository.



4. Discussion and Evaluation

The exercise described above—evaluating the OA data model through a number of domain-specific use cases - has enabled us to identify the model’s strengths and weaknesses.

The main strengths of the OA model that we have identified are as follows:

- *The OA model supports multiple targets, and each can be associated with a selector for specifying the segment of interest.* This has allowed us to create annotations that describe textual variation across multiple textual documents, to migrate a single annotation across multiple representations of a 3D museum object, and to attach a semantic tag to both accelerometry data streams and video that documents an animal’s movements—without extending the model or creating aggregate targets. By comparison, previous annotation data models, such as the Annotea ontology that underpins AO were designed with the assumption of a single target, so they provide no mechanism for associating an Annotea *Context* (used for describing the segment of a target resource) with a specific target.
- *Bodies and Targets can be any media type and can be located on any server.* This flexibility means that we can directly annotate digital resources that have been made available through online collections, such as those digitized and published by archives, museums, libraries and

publishers. It also allows us to create Bodies that are RDF, so that metadata properties associated with the Body can be stored separately rather than included in the annotation graph, making the provenance of the Body, the Target and the Annotation clear and explicit.

- *Because the OAC model is RDF-based, it is a trivial exercise to extend the model and include properties from existing domain-specific ontologies within the annotation graph.* For example, for the scholarly editions case study, we use custom properties to link target documents to FRBR entities, allowing us to query and retrieve annotations across multiple versions of the same FRBR expression or work.

The main weaknesses of the OA model that we have identified are:

- *The relative complexity of the OA model* for basic use cases. It is possible that many developers who want to implement simple tagging of Web pages or whole digital resources, will find the OA model too complex for their needs and hence it will not be widely adopted. (On the other hand it is the flexibility afforded by such complexity that enables the OA model to represent complex scholarly annotation use cases, that simpler models like Annotea are not capable of supporting.)
- *The ambiguity that exists within the model.* For example, there are multiple ways to represent resource segments. Such ambiguity increases the development effort required to produce tools that fully implement the model. For example, for annotations on part of an image, the image segment could be specified: using an *SVGSelector* (with a *constrains* relationship to the image URI); using a media fragment identifier in the target URI (with an *isPartOf* relationship to the image URI); or using a media fragment expressed as a *oa:FragmentSelector*. In all three cases there is no direct link from the annotation object to the image URI, so a query to retrieve all annotations on a given image must examine the target URI (for “whole of image” annotations), as well as URIs related via properties.
- *The explosion of URIs.* The Semantic Web/Linked Open Data approach adopted by OA, recommends that every resource (Bodies, Annotations, Targets, Selectors, States, Styles) is accessible on the Web via a persistent URI. Annotation authoring and management tools will need to create, track and manage large numbers of URIs.
- *The current OA model does not support multiple Bodies*—only multiple Targets. A common situation that we found in our case studies is the need to attach both multiple semantic tags (or keywords), as well as a textual description to a resource, within a single annotation event. The current model recommends that you create multiple separate annotations. But this creates redundancy and also does not accurately reflect the annotation event. An alternative approach is to attach the textual description as an inline Body and to use the “hasSemanticTag” property to attach the semantic tags. As discussed in the next bullet point, we don’t support the “hasSemanticTag” property that is currently included in the Extension ontology as it is inconsistent with the existing class and property hierarchy.
- *Lack of Support for multiple Semantic Tags.* A *SemanticTag* should ideally be defined as a subclass of the *oa:Annotation* class in the Extensions ontology. In addition, the Core OA ontology should allow multiple Bodies to be defined. These two changes would enable the

model to support the use case that we describe above, and simultaneously maintain a well-structured and consistent ontology.

- *Performance issues when Querying.* The complexity of the OA model may lead to slow performance when querying and retrieving on annotation fields that are buried a number of levels down the Annotation graph. For example, the most common query is on Targets e.g., “give me all of the annotations on this Target resource”. The Target resource may be the object of the *hasTarget* or *hasSource* properties, or a selection that is retrieved by applying the *Selector* to *hasSource*, or part of an *Aggregated Target*. Resolving all of these alternatives to match the Target URI and retrieve and display the relevant annotations, may require optimization to improve performance.
- *Lack of standards for specifying segments of resources.* The ability to use segments or fragments of resources as Bodies or Targets, is extremely useful. However the interoperability of this aspect of the model is limited by a lack of standards across communities for describing/identifying parts of things e.g., text segments across document formats. The W3C Media Fragments Working group recently published a Proposed Recommendation which can be applied to multimedia resources (images, video, audio)—but there is a real need for communities to agree on schemas/mechanism for selectors on textual resources, maps, timelines and 3D objects.

5. Conclusions

In this paper we firstly describe the Open Annotation (OA) model that has recently been developed through the W3C Open Annotation Community Group by aligning the OAC and AO data models. We then apply this model to a number of use cases that have arisen within collaborative projects conducted through the UQ ITEE eResearch Lab. We have demonstrated how the OA model can be applied and extended in each use case, to support scholarly annotation practices ranging from basic attachment of comments and tags to digital resources, through to relating multiple target resources, and data annotation. We have also illustrated how the OA model has enabled us to create common backend infrastructure for storing, indexing and querying annotations across applications and authoring clients.

We have identified and discussed some of the outstanding issues that need to be addressed in order to improve the interoperability capabilities of this approach.

By exposing annotations to the Semantic Web as Linked Open Data, the Open Annotation approach enables robust machine-to-machine interactions and automated analysis, aggregation and reasoning over distributed annotations and annotated resources. Moreover, this annotation environment that will allow scholars and tool-builders to leverage traditional models of scholarly annotation, while simultaneously enabling the evolution of these models and tools to make the most of the potential offered by the Semantic Web and Linked Data environments.

Acknowledgments

We gratefully acknowledge the contributions to this paper and to the annotation cases described within from the following individuals: Roger Osborne and Paul Eggert (scholarly editions); Chih-Hao Yu (3D museum objects); and Lianli Gao (sensor data streams). We also wish to thank the

Andrew W. Mellon Foundation for their generous funding of the OAC project and our collaborators from the Open Annotation Collaboration (in particular Tim Cole, Herbert Van de Sompel, Rob Sanderson) and the W3C Open Annotation Community Group.

References

1. Hunter, J. Collaborative semantic tagging and annotation systems. *Ann. Rev. Inf. Sci. Technol.* **2009**, *43*, 187–239.
2. Sanderson, R.; van de Sompel, H. Open annotation: Beta data model guide. *Open Annot. Collab.* **2011**. Available online: <http://www.openannotation.org/spec/beta/> (accessed on 12 April 2012).
3. Kahan, J.; Koivunen, M.R.; Prud'Hommeaux, E.; Swick, R.R. Annotea: An open RDF infrastructure for shared Web annotations. *Comput. Netw.* **2002**, *39*, 589–608.
4. Agosti, M.; Ferro, N. A formal model of annotations of digital content. *ACM Trans. Inf. Syst.* **2007**, *26*. doi: 10.1145/1292591.1292594.
5. Boot, P. A SANE approach to annotation in the digital edition. *Jahrb. Computerphilologie* **2006**, *8*, 7–28.
6. Bateman, S.; Farzan, R.; Brusilovsky, P.; McCalla, G. OATS: The open annotation and tagging system. In *Proceedings of the Third Annual International Scientific Conference of the Learning Object Repository Research Network*, Montreal, Canada, 8–10 November 2006.
7. Ciccarese, P.; Ocana, M.; Garcia Castro, L.J.; Das, S.; Clark, T. An open annotation ontology for science on web 3.0. *J. biomed. Semant.* **2011**, *2*, S4:1–S4:24.
8. Sanderson, R.; Ciccarese, P.; van de Sompel, H. Open Annotation draft data model. *Open Annot. Collab.* **2012**. Available online: <http://www.openannotation.org/spec/core/> (accessed on 24 August 2012).
9. Koch, J.; Velasco, C.A.; Ackermann, P. *Representing Content in RDF 1.0*; W3C Working Draft 10 May 2011; W3C: Cambridge, MA, USA, 2011. Available online: <http://www.w3.org/TR/Content-in-RDF10/> (accessed on 24 August 2012).
10. The Core Open Annotation Namespace. Available online: <http://www.w3.org/ns/openannotation/core/> (accessed on 24 August 2012).
11. The Extensions Namespace. Available online: <http://www.w3.org/ns/openannotation/extension/> (accessed on 24 August 2012).
12. International Federation of Library Associations and Institutions (IFLA). *Functional Requirements for Bibliographic Records*; IFLA: Hague, the Netherlands, 1998.
13. AustESE Project. Available online: <http://austese.net/> (accessed on 24 August 2012).
14. Open Archives Initiative—Object Reuse and Exchange (OAI-ORE). *ORE User Guide—Resource Map Implementation in RDF/XML*; OAI: New York, NY, USA, 2008. Available online: <http://www.openarchives.org/ore/1.0/rdfxml> (accessed on 24 August 2012).
15. Gerber, A.; Hunter, J. Compound Object authoring and publishing tool for literary scholars based on the IFLA-FRBR. *Int. J. Digit. Curation* **2009**, *4*, 28–42.
16. Eggert, P.; McQuilton, J.; Lee, D.; Crowley, J. Ned Kelly's Jerilderie Letter: The JITM Worksite. Available online: http://web.srv.adfa.edu.au/JITM/JL/Annotation_Viewer.html (accessed on 24 August 2012).

17. Yu, C.H.; Groza, T.; Hunter, J. High Speed capture, retrieval and rendering of segment-based annotations on 3D Museum objects. *Lect. Notes Comput. Sci.* **2011**, *7008*, 5–15.
18. W3C Media Fragments Working Group. *Media Fragments URI 1.0 (Basic)*; W3C Proposed Recommendation; W3C: Cambridge, MA, USA, 2012. Available online: <http://www.w3.org/TR/2012/PR-media-frags-20120315/> (accessed on 24 August 2012).
19. OzTrack Project. Available online: <http://www.oztrack.org/> (accessed on 24 August 2012).
20. Flot—Attractive Javascript Plotting for jQuery. Available online: <http://code.google.com/p/flot/> (accessed on 24 August 2012).
21. VideoJS—The Open Source HTML5 Video Player. Available online: <http://videojs.com/> (accessed on 24 August 2012).
22. Lorestore on GitHub Home Page. Available online: <https://github.com/uq-ereseach/lorestore/> (accessed on 24 August 2012).

© 2012 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).