

Beyond Annotea – Open Annotations

Stephen Crawley¹, Ron Chernich², Jane Hunter²

eResearch Lab, University of Queensland, Brisbane, Australia

¹<uqscrawl@uq.edu.au>, ²<chernich@itee.uq.edu.au>, ³<jane@itee.uq.edu.au>.

INTRODUCTION

The most popular distributed web annotation tools and services are based on proprietary protocols. Whilst a standards-based approach is preferable for a variety of reasons, the fact is that there is no standard for annotations. The closest that we currently have is the W3C Annotea Protocol document [1]. However, this is a draft document produced by a W3C working group that has never been formally endorsed or widely implemented. It does not have the standing of a W3C recommendation (or even a draft recommendation), and it does not have the degree of precision or completeness that you would expect of a standards document. In addition, the Annotea document does not adequately address the following areas:

- There is no clear model of what an annotation is, and how it relates to the resource or resources being annotated.
- Annotations of resource fragments (e.g. paragraphs in web pages) are not adequately addressed.
- The document does not *specify* the schema for basic annotations and replies, and provides no framework or guidelines for domain specific annotation schemas.

The OpenAnnotation Collaboration [2] is developing over-arching conceptual model for annotations that is based on OAI-ORE aggregations, and may also address the annotation schema and resource fragment issues.

ANNOTEA IMPLEMENTATION ISSUES

Our experience with implementing Annotea is that the protocol is problematic in a number of areas. For example:

- There is no good way to identify the creator / owner of an annotation, and no recommended way to implement authorization or access control between Annotea clients and servers.
- There is no consideration of the security implications of displaying HTML annotation and reply bodies created by one user in another user's web browser.
- The Annotea document does not say what a server or client should do with RDF properties that it does not recognize. Should they be retained? Should they be dropped? Is it acceptable to “fix” them?
- The Annotea document is not clear about where the boundary of an annotation is from the perspective of a GET request. Is it simply the triples that have the annotation's URI as subject? Does it include dependent blank nodes? Does it include other non-blank nodes referenced by annotation properties?

In implementing Danno annotation server and Dannotate annotation tool [3][4] for the Atlas of Living Australia [5], we have developed common-sense solutions to these and other problems with the Annotea document. However, it is a fact that interoperability with other Annotea clients and servers can be problematic. This is not helped by the fact that some clients and servers have been implemented against the W3C server rather than the protocol document, and the former diverges from the later in some areas.

ANNOTEA AND TRIPLE-STORES

Assuming an adequate knowledge of semantic web technologies, it is fairly simple to build an Annotea-based server, save all of the annotations to an off-the-shelf triple store and map Annotea queries onto SPARQL queries. This is roughly what we did to implement Danno. Annotea queries and Danno-specific queries are internally mapped to SPARQL, and triple-store operations (queries and updates) are performed using an abstraction layer that hides each store's Java APIs. The performance is adequate for a relatively low traffic site; i.e. ~100 queries per second using Sesame, Jena SDB or Jena RDB. However there are a number of outstanding issues, described below.

The main lessons we have learned from implementing Danno are as follows:

- Using an off-the-shelf triple-store for storing annotations is not conducive to fast retrieval of annotations. Supporting blank-nodes inevitably results in extra queries to retrieve each annotation's nodes. From a performance perspective, it would be better to store the annotations as XML/RDF blobs in a relational database.
- Triple-stores and SPARQL do not really add much when implementing Annotea-style queries. Even allowing for site-specific extensions, the limited repertoire of the queries used by a typical annotation tool can easily be mapped to SQL queries against a relational database. It would also be a bad idea to expose a SPARQL query interface to users because of the difficulty of managing resource usage.
- Supporting multiple triple-store back-ends is a difficult proposition. Firstly there are no standard Java APIs for triple-stores. Secondly, transaction support tends to be patchy across the different triple-stores, and in some cases is non-existent. Finally, different triple-store implementations have different ideas about what constitutes valid RDF, especially URIs.

SCALABILITY

For the current design and implementation of the Danno server, we have assumed an annotation base and community of users small enough to be supported by a single annotation server with a single back-end triple-store. However, we have also tried to be mindful of the problems of a large-scale deployment of an annotation service. For normal Annotea usage patterns, there are three scaling variables; the number of annotated resources, the number of annotations and replies per resource, and the number of simultaneous users. So for example, a hypothetical implementation of Danno for use by Australian academics might ultimately need to handle millions of annotated resources, each with hundreds of annotations and replies per resource and thousands of simultaneous users. This is well beyond Danno's current capabilities.

Fortunately, there is an obvious strategy for scaling up – **distributed annotation servers**. The two main queries supported by Annotea are “find annotations for a given target resource URI” and “find the replies rooted in this annotation”. This suggests a straight-forward partitioning where annotations are distributed to different annotation servers based on a prefix or hash of the target resource URI, and replies are similarly distributed by annotation URI. A front end server can then direct Annotea requests to the relevant server based simply on the request URI and query parameters. Queries that cut across the partitioning (for example, “find all annotations by Donald Knuth”) could be handled by maintain separate annotation indexes keyed by the relevant annotation properties (in this case “dc:creator”).

Some annotation tools repeatedly check for annotations on the same resource URI. For instance, when “live updates” are enabled, Dannotate will poll the annotation server every 30 seconds for annotation updates relevant to the pages the user is looking at. Any updates are then applied to the displayed pages automatically. While users find this mode of operation very attractive, it places considerable load on the annotation server. One possible solution is to feed annotation and reply life-cycle events into an internal publish-subscribe system. These events would be read and accumulated by a number of “live update” servers which the user’s browser could poll for updates.

BROADENING THE SCOPE OF ANNOTEA

The original conception of Annotea was as a way of annotating web pages. Since then, the Annotea model has been adapted to wider tasks such as annotation of images, audio and video resources [6] and the representation of bookmarks [7]. We are currently working on further forms of annotation:

- In the ALA, the target of an annotation is typically (for example) a species or occurrence record that may be *displayed* in a web page. The idea is that the annotations for a record could be displayed or created in any context in which the record is displayed. This means that a free-standing annotation tool (like Dannotate) needs know what regions of a web page are appropriate to annotate, what kind of annotations to allow, and how to map those regions to annotation target URIs. This knowledge could be supplied in a variety of ways.
- We are currently experimenting with displaying annotations for geo-referenced observations using GoogleEarth.
- We are also investigating support for the Open Annotations Collaboration model.

We are also considering an overhaul to the Danno code-base to generalize from annotations to “objects” comprised of a set of related RDF triples. As an example, this would allow LORE [8] to use Danno as its repository of compound object descriptions. Among other things, the enhanced Danno server would need to implement a mechanism for determining an object triple set's boundary and its schema, and a mechanism for specifying the queries that an HTTP client would use to retrieve objects. Given such extensions, Annotea will facilitate the incorporation of web-based annotations across many different applications into the Web of Linked Data.

ACKNOWLEDGEMENTS

This work is part of a National eResearch Architecture Taskforce (NeAT) project, supported by the Australian National Data Service (ANDS) through the Education Investment Fund (EIF) Super Science Initiative, and the Australian Research Collaboration Service (ARCS) through the National Collaborative Research Infrastructure Strategy Program.

REFERENCES

1. Swick, R., Prud'hommeaux, E., Koivunen, M., Kahan, J., *Annotea Protocols*. W3C. Available as <http://www.w3.org/2002/12/AnnoteaProtocol-20021219>, accessed 30 June 2010.
2. *Open Annotation Collaboration*. Available as <http://www.openannotation.org/>, accessed 30 June 2010.
3. Chernich, R., Crawley, S., Hunter J., *Universal Collaborative Annotations with Thin Clients – Supporting User Feedback to the Atlas of Living Australia*, eResearch Australasia 2009.
4. *Danno / Dannotate Overview*. (Maven project website). Available as <http://metadata.net/sites/danno/>, accessed 30 June 2010.
5. *The Atlas of Living Australia*. Available as <http://www.ala.org.au/>, accessed 30 June 2010.
6. Schroeter, R., Hunter, J., Kosovic, D., *FilmEd - Collaborative Video Indexing, Annotation and Discussion Tools Over Broadband Networks*. Proceedings of the Multimedia Modelling Conference 2004. Brisbane, Australia. January 2004. pp 346 - 353.
7. Koivunen, M., Swick, R., Kahan, J., Prud'hommeaux, E., *An Annotea Bookmark Schema*. W3C. Available as <http://www.w3.org/2003/07/Annotea/BookmarkSchema-20030707>, accessed 30 June 2010.
8. Gerber, A., Hunter, J. *LORE: A Compound Object Authoring and Publishing Tool for Literary Scholars*, Digital Humanities 09. University of Maryland, USA, June 22 - 25, 2009.