Supporting Scholarly Web Annotation via RDFa and Annotation Ontologies

Marijn Koolen
Huygens ING, Royal Netherlands Academy for Arts and Sciences
marijn.koolen@huygens.knaw.nl

Jaap Blom
Netherlands Institute for Sound and Vision
jblom@beeldengeluid.nl

Peter Boot
Huygens ING, Royal Netherlands Academy for Arts and Sciences
peter.boot@huygens.knaw.nl

Ronald Haentjens Dekker
Huygens ING, Royal Netherlands Academy for Arts and Sciences
ronald.dekker@huygens.knaw.nl

Liliana Melgar-Estrada
University of Amsterdam/Netherlands Institute for Sound and Vision
melgar@uva.nl

ABSTRACT

Annotation is essential to many research disciplines. With more and more resources made accessible via the web, there is an urgent need to provide web-centric support for scholars to make, organize, share and analyze annotations across heterogeneous, distributed and complex digital resources. Most browser-based annotation tools access the HTML presentation layer used to display resources, which makes annotations dependent on a specific view and has little meaningful connection to the rich semantic structure of the underlying resource. This paper presents an approach to scholarly web annotation in which annotation clients exploit semantic information about resources embedded in RDFa, so they can 1) understand how different parts of a resource are semantically related to each other, 2) provide references to the annotated object that are independent of a specific web-based view, 3) aggregate annotations at different levels of a resource, and 4) provide annotation suggestions based on what part of a resource is being annotated. The main implications of this approach are that 1) scholarly annotations tools can work on any media type, 2) content providers have a low barrier to adopt this approach and make their resources annotatable and thereby more useful for researchers, and 3) annotation tools can more easily inter-operate and exchange annotations, such that researchers can make, organize, query and analyze their annotations across collections, media types, content providers, as well as across different web representations of same content.

KEYWORDS

Scholarly annotation, W3C Web Annotation model, RDFa, Digital Humanities

ACM Reference format:
Marijn Koolen, Jaap Blom, Peter Boot, Ronald Haentjens Dekker, and Liliana Melgar-Estrada. 2017. Supporting Scholarly Web Annotation via RDFa and Annotation Ontologies. In *Proceedings of SEMANTiCS Conference, Amsterdam, Netherlands, September 2017 (SEMANTiCS'17),* 8 pages.
https://doi.org/...

SEMANTiCS'17, September 2017, Amsterdam, Netherlands

ACM ISBN ......$...
https://doi.org/...

1 INTRODUCTION

Annotation is a fundamental activity in many research areas [10] and especially in the humanities [26, 28]. Through large-scale digitization efforts and developments of digital research infrastructures, many researchers access scholarly resources via the Web, creating an urgent need for appropriate support for scholarly web annotation. The use cases of scholarly annotation are highly complex, covering different forms of annotation across heterogeneous, richly structured and highly distributed resources of various media types such as historical archives of newspapers, television and radio broadcast, books, pamphlets and correspondences. Do to copyright restrictions, access to these resources is often restricted to an online environment created by content providers, which means support for scholarly annotation has to be brought to these environments. Beyond the creation of annotations, there is a need to support organizing and analyzing annotations based on the (parts of) the resource that are annotated. For instance, historians may want to compare sentiment-themed tags on newspaper articles with those on personal correspondences to investigate differences between public and private communications on the topic of migration.

There is increasing interest in web annotation [1, 10, 12, 14, 23] and development of annotation tools supporting different forms of annotation, such as commenting, tagging and classifying and linking to vocabularies and ontologies. Most browser-based annotation tools access the HTML presentation layer used to display resources, which makes annotations dependent on a specific view and has little meaningful connection to the rich semantic structure of the underlying resource.

For an approach to scholarly web annotation to be widely adopted, we argue that it should meet requirements for both scholars and content providers. One aspect of this is that annotation functionality should be able to operate on the rich structural information about resources offered by content providers such that researchers can select, filter and analyze annotations based on information about the resources they annotate. For content providers the integration of annotation functionality should require minimal effort.

For scholars the functionality should be open and flexible, resulting in annotations that can be easily exported, shared, combined and analyzed, while retaining information about their provenance.

This paper presents an approach to scholarly web annotation in which annotation clients exploit semantic information about resources embedded in RDFa, so they can 1) understand how different parts of a resource are semantically related to each other, 2) provide references to the annotated object that are independent of a specific web-based view, 3) aggregate annotations at different levels of a resource, and 4) provide annotation suggestions based on what part of a resource is being annotated. We are implementing a prototype in the Dutch national research infrastructure CLARIAH[1] to support scholarly annotation for various collections of text, image, audio and video resources.

In earlier work [5] we have argued that for supporting scholarly annotation, detailed information is needed about the annotated object, the annotator and the annotation itself. In this paper we focus on the annotated object. The main implications of this approach are that 1) scholarly annotations tools can work on any media type, 2) content providers have a low barrier to adopt this approach and make their resources annotatable and thereby more useful for researchers, and 3) annotation tools can more easily inter-operate and exchange annotations, such that researchers can make, organize, query and analyze their annotations across collections, media types, content providers, as well as across different web representations of same content.

2 RELATED WORK

2.1 Scholarly Annotation

Winget [30] observes that annotations have existed since there is text to annotate, because annotating is a way to understand, interpret, and create new knowledge. The most common annotation scenario in this sense is reading [27], since readers become annotators as they "wander, collect, organize, interpret, mark in, and mark on what they gather" [15]. The Annotating All Knowledge Coalition[2] promotes the creation of a "conversation layer" over the Web, where annotation is the unit of the conversation. In the humanities domain annotation has been also identified as one of the main scholarly primitives [22, 26]. The W3C Web Annotation Data Model [24] lists thirteen types of annotations as motivations, including *classifying*, *commenting*, *editing*, *highlighting*, *linking* and *tagging*, which roughly correspond to activities identified in Palmer et al. [22] and in the Taxonomy of Digital Research Activities in the Humanities.[3] Marshall [16] also observed that annotations vary according to different dimensions, such as the level of formality, intelligibility (from tacit to explicit), their function during reading, their intensity, permanence, privacy, and audience, and that they are a key to grow the hypertext "ecology", since they "promote an accretion of both structure and content" [16]. Still, in relation to formality, there seems to be a difference between the "lightweight" personal annotations that Marshal discusses, and scholarly annotations, which benefit from a higher level of structure [13] that is offered by an increasing number of semantic annotation tools.

[1]Common Lab Research Infrastructure for the Arts and Humanities https://www.clariah.nl/en/

[2]https://hypothes.is/annotating-all-knowledge/

[3]TaDiRAH, see http://tadirah.dariah.eu/vocab/index.php

2.2 Semantic Annotation Tools

There are several types of annotation tools ranging from bookmarking web services to more complete stand-alone qualitative data analysis software [18]. Annotation tools or services that benefit from the web infrastructure are relatively recent. In 2000, Unsworth [26] commented that "shared annotation is, for all scholarly intents and purposes, impossible on the Web". In 2012, [21] argued that it was "a good time to resolve this situation" .

In line with [24] we refer to the object that is annotated as the annotation *target*, the content of the annotation as the annotation *body* and who or what creates the annotation as the annotation *creator*. Currently, several tools and initiatives are working on the principles of web protocols and standards to support annotation, such as the Annotating All Knowledge Coalition, which aims to build annotation facilities "in the very fabric of the web". Examples of this approach are Apache Annotator[4] and the Open Knowledge Foundation project Annotator, [5] an open-source JavaScript library for building annotation applications in browsers. Applications using Annotator include Hyphothes.is, [6], and Annotation studio. [7]A step further in building annotation tooling based on web standards is the Pund.it web annotation suite[8], which builds on semantic web technologies and the Web Annotation Data Model. Pund.it was used as the main application for the *Burckhard's Correspondence Project*,[9] in which scholars could enrich the letters by adding semantic annotations (annotation bodies) to mentions of entities such as persons, places and works (annotation targets) [12].

The Domeo Annotation Toolkit[10] is another JavaScript library using the Web Annotation standard, that content providers can easily include in their resource pages. Although developed for the biomedical domain, it is also applicable to other areas. It enables manual, automatic, and semi-automatic adding labels (bodies) to text and images (targets), with an emphasis on structured, semantic annotations [8], meaning that content is identified via live ontologies. Currently, Domeo is part of the Annotopia initiative for "desiloing annotation" [9]. The A4 system (Annotations Anywhere, Annotations Anytime) system [25] implements fine-grained annotation of text by guiding the user through the annotation process via semantic templates which use the CIDOC CMR ontology model.

Most of these applications are focused on text annotation [11, 20], though there are initiatives and applications for semantic image and video annotation. The ArtTube[11] and LinkedTV projects [3] support semantic web annotation of video using modifiable semantic templates.

The examples mentioned above provide advanced support for using semantics in the *body* of web annotations, but they are limited in using semantic information regarding the *target*.

[4]http://annotator.apache.org/

[5]http://okfnlabs.org/projects/annotator/

[6]https://web.hypothes.is/

[7]http://www.annotationstudio.org/

[8]http://thepund.it/

[9]http://burckhardtsource.org/

[10]http://annotationframework.org/

[11]http://www.arttube.nl/

3 SUPPORTING SCHOLARLY WEB ANNOTATION

In this section we describe use cases for scholarly web annotation, derive requirements for developing annotation functionality to support it, then describe the overall architecture and the responsibilities and characteristics of different components of the architecture.

3.1 Scholarly Use cases and Web Annotations

Scholarly use cases for annotation vary depending on discipline, research focus or goals, and media type [12]. The three scholarly use cases below originate from three different disciplines. They illustrate the specific scholarly need for annotations that have meaningful and stable references to the structure of the annotated resources:

- A textual scholar wants to annotate various parts of a digital scholarly edition of the *Correspondence of Vincent van Gogh*[13], such that the annotations can be used and interpreted independent from the way the edition is displayed [4], and be selected based on which part of the letters they target (e.g. all annotations on the entire resources, or only the annotations on specific parts, e.g. on the original texts, on translations, or on metadata).
- A media scholar wants to investigate the representation of ethnic minorities on Dutch news and current affairs in the 1990s. For this analysis, the scholar wants to select, annotate, and compare television and radio programs, as well as newspapers and entertainment magazines. The focus of this scholar is both on media discourse (verbal) as well as on visual representations depending on genre. Thus, the scholar needs to annotate fragments of the individual episodes of a program or article and aggregate annotations depending on the type and genre of the annotation target (e.g. all annotations of entertainment media or all annotations of news items).[14]
- A historian of science wants to investigate uses of medical drug components in the domains of science, commerce and public debate in the period 1500-1800 and wants to annotate mentions of medical drug components, and related entities including people, organisations, locations, actions and objects in digitized newspaper articles, books, historical inventories and trading records, and early modern archives. During the research process, the historian wants to query the annotation server for various subsets of the annotations for analysis, aggregating the annotations on e.g. the type of document they target (trading records, scientific correspondences, personal letters).

Such use cases illustrate a limitation of annotation tools that use HTML-based selectors for scholarly analysis, such as CSS and XPath selectors[15]: annotations in this format can be easily retrieved or sorted in categories based on the annotation *body*, such as tags or classification terms. However, there are limited ways of meaningfully selecting and analyzing annotations by different *target* types when the target is identified by a URL. It is not clear how a researcher can easily retrieve annotations targeting (sub-resources of) resources by genre or type (e.g., news broadcasts or letters), or based on their internal structure (e.g., chapter, sequence) as this can rarely be derived from the HTML layer.

3.2 Requirements and Characteristics

We want an approach to scholarly annotation that addresses the needs of scholars while being easy for resource providers to participate in. To address the needs of scholars, the use cases above indicate the approach should support 1) meaningful targeting, i.e. being able to distinguish between different types of targeted resources, 2) mixed media annotation, 3) different annotation tasks (the body of an annotation) and 4) contextualisation (creator context and motivation).

To encourage resource providers to participate and make annotation of their resources possible, we aim for a low technical threshold. Similar to several of the tools mentioned in Section 2.2, The annotation client is offered as a highly configurable JavaScript library that can be easily injected in a resource viewer, and a generic annotation ontology is provided which can be used directly or extended, to describe resources and their annotatable components. This does require that resource providers create stable identifiers for annotatable parts.

3.2.1 Requirements. As described in earlier work [5], annotation support for the use cases above leads to the following technical requirements:

- (R1) the resource describes itself and its structure to the annotation tool, and provides suitable labels for the annotatable objects;
- (R2) the resource provider can suggest annotation types for the annotatable objects;
- (R3) the effort to integrate annotation functionality for existing resources is minimal;
- (R4) the annotation targets are durable URIs and not formulated in terms of HTML structure;
- (R5) URIs should be treated as opaque (i.e., we shouldn't try to guess the relations between the annotated components based on their URIs);
- (R6) URIs should be canonical; and lastly
- (R7) the annotation tool is generic, but able to handle the created annotations with awareness of the structure that they apply to (it can e.g. return aggregated annotations).

3.3 Architecture and Responsibilities

The content provider offers resources via a *resource server*, which are displayed in a *resource viewer* offered by either the content provider or by an external site. The annotation functionality is offered through an *annotation client* that is incorporated in the *resource viewer* and that communicates with an *annotation server* that stores and gives access to the annotations.

- The *resource server*
  - serves resource data to a resource viewer,

[12] The Annotating All Knowledge Coalition lists as use cases: peer review, personal note taking, post-publication discussion, among others in reading and digital document development. See also the DARIAH-EU Working Group Digital Annotations use case survey [27]

[13] See http://vangoghletters.org/vg/

[14] See other use cases for annotation in media studies in Melgar et al. [17], and Bron et al. [6]

[15] https://www.w3.org/TR/annotation-model/#selectors

  - determines what parts of the resource can be annotated, specified in an annotation ontology[16],and any extension for a specific domain or edition, e.g an extension for the Van Gogh Correspondence.[17].
  - provides URIs for annotatable sub-resources so they can be referred to in annotation targets.
  - optionally, provides a list of allowed, preferred and/or suggested annotation types.
- The *resource viewer*
  - fetches resource data from the *resource server*,
  - embeds descriptive information about the resource through RDFa properties in the HTML representation,
  - embeds and configures the annotation client,
- The *annotation client*
  - runs in the browser and is embedded in *resource viewer*,
  - uses the resource structure and ontology in handling annotations,
  - exchanges annotations with the annotation server.
- The *annotation server*
  - stores annotations,
  - returns annotations for queries based on resource URIs as targets of annotations, or other properties of annotations such as identifier, creator, body type.

The resource provider may wish to maintain their own annotation server or configure the annotation client to communicate with an external annotation server. To handle queries for annotations based on resource URIs requires the annotation server to identify annotations that target the resource itself or any of its sub-resources. In other words, it needs access to information about how resource URIs are structured into hierarchies.

We describe how our annotation approach addresses the requirements for self-describing resources (requirements R1-R6) in Section 4. We describe how the annotation sever can use resource structure information for querying and aggregating annotations (requirement R7) in Section 5.

4 DESCRIBING RESOURCES THROUGH RDFA

In this section we discuss our proposed annotation approach where content providers can enrich the browser-based presentation of scholarly resources via RDFa, and how annotation client and server can exploit this information to support the use cases and requirements described in Section 3.

The RDFa framework[18] was introduced to embed rich metadata in Web documents. It offers a way for content providers to enrich the presentation of their resources on the Web with machine-readable RDF expressions. RDFa is one of the ways in which websites can describe themselves for search engines to interpret the various parts of a web page. Search engine companies Bing, Google and Yahoo launched schema.org[19] to provide a common set of schemas for this purpose. The RDFa framework can also be used for describing resources for annotation purposes. We illustrate our approach via an example of annotating a richly structured letter from the *Van Gogh Correspondence*. The letter as a creative work consists of a transcribed version of the original Dutch text, image scans of the letter, metadata regarding sender, receiver and sending date, as well as footnotes added by the editors and an English translation of the letter. In order to be able to describe the various components of the web page to the annotation client, we introduce an ontology that describes the annotatable things represented on the web page. The ontology provides the basics for those resource providers who do not want to invest in a specialized ontology describing their resources. More ambitious resource providers always have the option to extend the ontology with more specific classes and properties.

```
<div vocab="http://boot.huygens.knaw.nl/annotate/vangoghontology.ttl#" typeof="Letter"
about="urn:vangogh:let001">
        <p resource="urn:vangogh:correspondence" typeof="Correspondence" property="isPartOf">
                <a href="http://vangoghletters.org/vg/">Van Gogh Letters</a></p>
        <h2>To Theo van Gogh. The Hague, Sunday, 29 September 1872.</h2>
        <div class="metadata">
                <span typeof="Sender" property="hasMetadataItem"
resource="urn:vangogh:let001.sender">Vincent van Gogh</span>
                <span typeof="Receiver" property="hasMetadataItem"
resource="urn:vangogh:let001.receiver">Theo van Gogh</span>
                <span typeof="Date" property="hasMetadataItem"
resource="urn:vangogh:let001.date">Sunday, 29 September 1872</span>
        </div>
        <div class="transcription">
                <p property="hasPart" typeof="ParagraphInLetter"
resource="urn:vangogh:let001:p.1">Den Haag, 29 september 1872.</p>
                <p property="hasPart" typeof="ParagraphInLetter"
resource="urn:vangogh:let001:p.2">Waarde Theo,</p>
        </div>
        <div class="transcription" typeof="CreativeWork Translation" property="hasEnrichment"
resource="urn:vangogh:let001.trans">
                <p property="hasPart" typeof="ParagraphInLetter"
resource="urn:vangogh:let001:translation:p.1">The Hague, 29 september 1872.</p>
                <p property="hasPart" typeof="ParagraphInLetter"
resource="urn:vangogh:let001:translation:p.2">Dear Theo,</p>
        </div>
</div>
```

Figure 1: A HTML representation of a letter by Vincent van Gogh with resource information embedded through RDFa.

Our approach is comparable to the approach taken in [29], which, however, limits itself to describing the main resource contained in a web page. Since we want to facilitate fine-grained annotations at the level of sub-resources, we label the components of the web page and their relations using the annotation ontology. The top class in the ontology is *AnnotatableThing*. In the example for the domain of scholarly editing that we have worked out, the underlying classes are *CreativeWork* (borrowed from schema.org), *TextBearer* and *Enrichment*. These three classes should cover most of the content present in any edition. In the case of the Van Gogh correspondence, both the correspondence and the individual letters are *CreativeWorks*, the pages that the letters are written on are *TextBearers*, and the editiorial enrichments, such as metadata, notes and translations are *Enrichments*. But for the Van Gogh edition we also developed a more specific ontology, where for instance *CreativeWork* has specializations *Correspondence*, *Letter* and *ParagraphinLetter*, and the various metadata fields are given in specializations of *Enrichment*.

A HTML representation of a fragment of the first letter is shown in Figure 1. The metadata elements and paragraphs of the original Dutch text and the English translation each have an identifier and information about type and relationship with the letter. We also include a class for web page content that is not to be annotated (not shown in the example). It can be used for example for boilerplate text surrounding the content of the page, or for the labels that identify the content.

[16]See a preliminary version for digital editions at http://boot.huygens.knaw.nl/annotate/genericontology.ttl
[17]See http://boot.huygens.knaw.nl/annotate/vangoghontology.ttl
[18]https://www.w3.org/TR/2015/NOTE-rdfa-primer-20150317/
[19]https://schema.org/

In order to facilitate aggregation, we need to define clearly what exactly counts as a sub-resource of a resource. An essential ingredient in the annotation ontology is therefore the definition of a generic 'includes'-property. A *hasPart* relation defines the object as a sub-resource of the subject. But a letter as a creative work can have both a textual and a material representation, that should count as its sub-resources. In the generic annotation ontology, we therefore use the 'includes' property for all hierarchical relations, with several properties as sub-properties to indicate various relations between resources and sub-resources, such as *isCarriedOn* (which connects the creative work and its material bearer), *hasPart* and *hasEnrichment*. There exist a number of ontologies that share concepts with this one (such as schema.org, mentioned above, and the FRBRoo ontology [2]). We reuse concepts from these ontologies wherever possible, but start from specific annotation-related concepts. In future work we will investigate extending the ontology with FRBRoo concepts.

One of the reasons why it is useful to describe the resource structure to the annotation tool, is that certain annotation types are suitable for certain resource types only. Annotation is not necessarily a completely user-driven activity. It can also be employed by resource owners as a technology in e.g. a crowdsourcing scenario [19]. In that case, resource owners might want to provide information about permitted or suggested annotation types, in order to be able to do further processing of the annotations. For instance, the resource owner could define an 'authority-link' annotation type for relating personal names to an entry in a suitable authority file. Or in a philosophical treatise users could be asked to relate paragraphs in the text to the concepts being discussed. In a review scenario, reviewers might be asked to judge a text on originality and readability using different annotation types. This is functionality that we have not as yet experimented with, but it is one of the motivations for the approach that we are taking towards structured annotation.

5 USING RESOURCE STRUCTURE

The requirement to support aggregation of annotations (R7 in Section 3) needs to be addressed in the communication between the components involved, that is, the resource provider and viewer and the annotation client and server. In this section we discuss how reasoning over the structure of resource can be handled to allow aggregation and querying of annotations. To discuss the issue of modeling and storing resource structural information, we use an example annotation on the letter in Figure 1, where the researcher annotates the second paragraph of the English translation of the letter, containing the text "Dear Theo" (Dutch original: "Waarde Theo"), with a classification label to indicate it as a salutation.

To allow aggregating annotations for the letter as a whole or for the translation, the annotation server must be able to reason over the structure of resources and sub-resources. For instance, annotations on the translation include annotations on the paragraphs of the translation, but not annotations on paragraphs of the original. But annotations on the letter as a whole includes annotations on both sets of paragraphs.

We compare three general approaches to representing the resource structure in the context of an annotation:

```
{
  "@context": "http://www.w3.org/ns/anno.jsonld",
  "created": "2017-05-10T08:50:49.055802+00:00",
  "body": [
    {
      "vocabulary": "DBpedia",
      "value": "Salutation",
      "purpose": "classifying",
      "id": "http://dbpedia.org/resource/Salutation"
    }
  ],
  "motivation": "classifying",
  "creator": "marijn",
  "type": "Annotation",
  "target": [
    {
      "id": "urn:vangogh:let001:translation:p.2",
      "type": ["Text", "ParagraphInLetter"]
    }
  ],
  "id": "urn:uuid:8f62be7d-de56-464b-8d9a-5fb0b69fc00b"
}
```

Figure 2: Example annotation of the salutation in a letter.

```
  "target": [
    {
      "source": "urn:vangogh:let001",
      "selector": {
        "conformsTo": "http://.../vangoghontology.json",
        "value": "urn:vangogh:let001.translation",
        "type": "FragmentSelector",
        "refinedBy": {
          "conformsTo": "http://.../vangoghontology.json",
          "value": "urn:vangogh:let001:translation.p2",
          "type": "FragmentSelector",
        }
      }
      "type": "Text"
    }
  ],
```

Figure 3: Example of embedding resource structural information in the annotation target.

(1) *Structure Embedded in Annotation*: embedding structural relation information in annotation targets.
(2) *Structure as Separate Annotation*: representing structural relation information as separate annotations.
(3) *Structure as Separate Model*: representing structural relation information in a separate data model.

5.1 Structure Embedded in Annotation

A simple approach is to store the entire resource structure in the annotation target, so that it can be retrieved for any resource in the hierarchy that contains the most specific resource, e.g. the second paragraph in the translation, as shown in Figure 3.

The advantages are:

- *simplicity*: It uses a single data structure for exchange.
- *interpretation*: The annotations contain a lot of information about the resource that aids interpretation out of the context of the resource itself.

The disadvantages are:

- *Conciseness and Redundancy*: annotations contain more structural information than necessary for many contexts. Multiple annotations on the same paragraph all contain information about its hierarchical relation with higher-level resources.
- *Richness*: only a very limited amount of structural information can be embedded in an annotation, which might not

```
{
  "@context": "http://www.w3.org/ns/anno.jsonld",
  "type": "Annotation",
  "created": "2017-05-10T08:50:49.055802+00:00",
  "body": [
    {
      "@context": "http://.../vangoghontology.jsonld",
      "value": "hasEnrichment"
    }
  ],
  "target": [
    {
      "id": "urn:vangogh:let001",
      "@context": "http://.../vangoghontology.jsonld",
      "type": "Letter"
    },
    {
      "id": "urn:vangogh:let001.translation",
      "@context": "http://.../vangoghontology.jsonld",
      "type": "Translation"
    }
  ],
  "id": "urn:uuid:8f62be7d-de56-464b-8d9a-5fb0b70fc00b"
}
```


Figure 4: Example of storing resource structural information as a separate annotation.

suffice for use cases that require more complex reasoning over resource structure.

- *Multiple parents*: annotations on the same resource made in different contexts, e.g. a translation as part of the original letter and as part of a collection of translations, show different parentage. Annotations made on the translation in the context of the collection of translations, will not be considered as annotations on the original letter.

5.2 Structure as Separate Annotation

It is possible to separate resource structure information from annotation information by representing them as separate data structures. One approach is to represent the structural relation between a resource and a sub-resource as a separate annotation, as shown in Figure 4. An additional annotation would be required to register that the second paragraph (*urn:vangogh:let001:translation:p.2* and the translation (*urn:vangogh:let001.translation* are connected via a *hasPart* relation.

The advantages are:

- *Simplicity*: all representations are W3C annotations.
- *Redundancy*: Each relation is stored only once, resulting in low redundancy compared to the annotation that embeds the resource structure, and only the structural relations of (sub-)resources are stored for the parts that are actually annotated.
- *Multiple parents*: Each hierarchical relationship is sent as separate annotation, so the server can easily traverse from different ancestors to the same descendant resource, as long as the relation between a sub-resource and each of its parents is registered.

The disadvantages are:

- *Conciseness*: each structural relation is sent as a separate representation which results in many annotations when a sub-resource in a deep hierarchy is annotated.
- *Separation of concerns*: Although the two types of information are sent in separate representations, they do not reflect the different natures of annotations and structural relations. The server has to store them as different types of annotations so it that knows that one is used to traverse a resource hierarchy for aggregation. Or it stores both as annotation but then, when receiving a request to send annotations on a resource, it has to determine which annotations refer to resource structure and which represent actual annotations.
- *Model fitness*: This may introduce ambiguity, such that it's not clear whether an annotation represents structure information (e.g. a translation belonging to the original) or an annotation to indicate that two resources are linked (e.g. linking a letter that mentions a painting to the identifier of that painting). This makes it problematic for the server to determine where to stop its traversal over resource structure annotations. It should not traverse to the painting and annotations on that painting. Solution could be to the use a special *motivation* value for structure annotations, or to extend W3C annotation model and create subclass for structure annotations.

5.3 Structure as Separate Model

A way to solve the problem of ambiguity is to represent structural information in a different data model, based on the ontology used to describe the resource. This requires the annotation server to either store information about how resources and sub-resources are structured or get this information from the content providers.

The latter is problematic in various ways. First, querying the server for annotations that target a particular resource type (e.g. *ParagraphInLetter*), would require the annotation server to send requests, for every resource specified in an annotation target, to their respective content providers, to check whether it is of the requested resource type of a sub-resource of a resource of the requested type. Second, if the content provider disappears, the annotation server can no longer reason over the resources targeted by the stored annotations. Third, this creates a higher threshold for content providers to participate as they need to provide an endpoint for the annotation server to access to this information.

For these reasons, we argue that the annotation server should store information about resource types and their relational structure. In this case, a choice has to be made on when the client sends structural information to the server and what structure information to send. A *lazy* client sends only structural relations between an annotated target and its ancestors when that annotation is made. A *pro-active* client sends the entire resource structure (i.e. the resource and all its sub-resources) when a new resource is loaded in the resource viewer in browser window. The resource structure is based on the RDFa elements embedded in the page. The example annotation is the same as in 2. All information regarding the relation between the annotated paragraph and the original letter, its translation and the larger correspondence should be handled separately in a structure-oriented data model.

A straightforward way for the client to communicate structural information about the resource is to send the RDFa information of a resource using the vocabulary that it is based on as context. In the case of the *van Gogh Correspondence*, this is the Van Gogh Correspondence extension of the annotation ontology described in Section 4: The annotation server can store all structural relations

```
{
  "@context": "http://.../vangoghontology.jsonld",
  "@type": "Letter",
  "id": "urn:vangogh:letter001",
  "hasMetadataItem": [
    {
      "@id": "urn:vangogh:letter001.sender",
      "@type": "Sender",
    },
    {
      "@id": "urn:vangogh:letter001.receiver",
      "@type": "Receiver",
    },
  ],
  "hasPart": [
    {
      "@id": "urn:vangogh:letter001:p.1",
      "@type": "ParagraphInLetter",
    },
    ...
  ],
  "hasEnrichment": [
    {
      "@id": "urn:vangogh:letter001.translation",
      "@type": ["CreativeWork", "Translation"],
      "hasPart": [
        ...
        {
          "@id": "urn:vangogh:letter001:translation:p.2",
          "@type": "ParagraphInLetter",
        },
        ...
      ],
    },
  ]
}
```


Figure 5: Example of storing resource structural information using a separate model.

including those between the original letter and its translation, and between the translation and its second paragraph. This allows traversal from any of these three resources to the annotation about the salutation.

The advantages are:

- *Simplicity*: the structural representation can lean entirely on the ontology used to describe the resource (which is the responsibility of the resource server).
- *Conciseness*: the structural representation only contains structural information.
- *Separation of concerns*: the hierarchical resource structure is modeled differently from the annotations, they can naturally be handled differently by the server.
- *Model fitness*: this makes full use of the annotation ontology and allows the server to use the same structure-related semantics as the client.
- *Redundancy*: The client first asks the server if it already knows about the resource. If not, the client sends the resource structure to the server upon parsing the resource in the browser window. Otherwise, no structural information needs to be sent.

The disadvantages are:

- *Redundancy*: information about resources that have not been annotated (yet) is sent to and stored by the annotation server.

5.4 Comparing Approaches

All three approaches have drawbacks, at least in that they duplicate resource structure from the content provider in the annotation server. We argue that communicating and storing resource structure *as a separate model* is the most flexible, and best fits the requirements derived from use cases of scholarly annotation. It clearly separates the two types of data and allows independent reasoning over structure. Extending the Web Annotation Data model with a special type to handle structure *as a separate annotation* is only a more elaborate alternative. Moreover, the *structure as a separate model* can easily be extended with additional information about resources for use cases where this is needed or useful. Note that it is still possible to generate (a special type of) W3C Web Annotations from the resource structure model for interoperability.

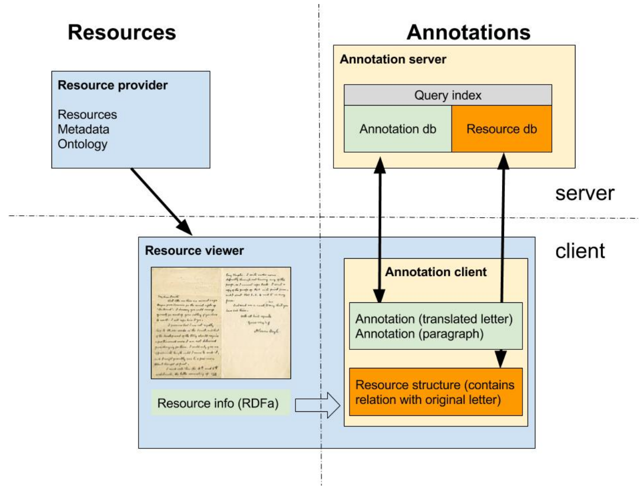

Figure 6: Architecture of the components involved and the data flow.

Our prototype implementation stores resource structure and annotations in a Neo4J graph database for efficient traversal either from resources to annotations or vice versa. The architecture and data flow for this approach is visualized in Figure 6. For the *Van Gogh Correspondence* we have an XSLT that transforms the TEI-encoded letters into an HTML presentation, and embeds information about annotatable parts as RDFa attributes. We are currently experimenting with generating RDFa-enriched detail views for the roughly 1 million TV and radio broadcasts in the archive of the Netherlands Institute for Sound and Vision.[20]

6 CONCLUSIONS

This paper introduced an approach to scholarly web annotation based on semantic descriptions of resources embedded in web pages via RDFa.

Using RDFa to semantically describe web pages in a machine-readable way offers many opportunities to enhance scholarly web annotation. With our proposed approach, content providers can choose which (collections of) resources to include and decide per collection or resource type in which ways they want to support annotation. The enhanced semantic information in annotation *targets* and the additional resource structural information captured in the annotation server offers scholars very flexible and complex

[20] http://www.beeldengeluid.nl/en

ways of aggregating, filtering and analyzing annotations. Although we illustrated this approach with an example of a digital edition of mainly textual resources, it generalizes to other media types. Archives of public radio and TV broadcasts can describe their resources, including audio and video streams, via RDFa. Segmented descriptions of audio and video streams can also be represented in HTML via the W3C SMIL standard [7], so that leaders, title sequences, shots and other parts can be labeled and selected as annotation targets. Researchers can easily combine annotations on different resource types, from different providers, stored in different annotation servers and use them for analysis.

We note that this approach is not meant as an alternative to the many existing tools and initiatives for web annotation, but as complementary functionality than can be incorporated to enhance support for reasoning over the annotatable elements of a resource, while maintaining current functionality to annotate web pages with RDFa information.

There are several steps still to be taken. One significant challenge that we have not yet addressed is dealing with changing or conflicting descriptions of the structure of a resource. Different content owners may offer the same material but have conflicting descriptions of the structure, or content providers can change the structure of resources, or the ontology used to describe it, based on changing interpretations and perspectives of curators (note that this is a problem for annotations in general). The benefit of capturing the structure of resources in the annotation server, either embedded in the annotation target, as a separate annotation or separate data model, is that allows reasoning over and interpreting resource structure at the time of creating the annotation. Versioning resource structure can be used to deal with changes in structure.

A second challenge is the persistence and logic of the provided identifiers of annotatable elements of resources, since this will impact for how long annotations will be properly linked and how reusable they are. We will also look at ways in which content providers can configure suggested annotation types.

Finally, we will investigate requirements for querying the annotation server. To support the types of queries required for searching, selecting and filtering annotations based on resource structure and annotation types, we have to investigate whether existing query languages suffice or have to be adapted.

REFERENCES

[1] Maristella Agosti, Giorgetta Bonfiglio-Dosio, and Nicola Ferro. 2007. A historical and contemporary study on annotations to derive key features for systems design. *International Journal on Digital Libraries* 8, 1 (2007), 1–19.

[2] Chryssoula Bekiari, Martin Doerr, Patrick Le Boeuf, and Pat Riva. 2015. Definition of FRBROO - A Conceptual Model for Bibliographic Information in Object-Oriented Formalism. (2015).

[3] Jaap Blom. 2014. *LinkedTV annotation tool, final release.* Technical Report Deliverable 1.5. https://www.slideshare.net/linkedtv/linked-tv-d15

[4] Peter Boot. Annotation in Digital Scholarly Editions (Flash talk). In *I Annotate 2016 summary of abstracts*.

[5] Peter Boot, Ronald Haentjens Dekker, Marijn Koolen, and Liliana Melgar. 2017. Facilitating Fine-grained Open Annotations of Scholarly Sources. In *Digital Humanities 2017.* To appear.

[6] Marc Bron, Jasmijn Van Gorp, and Maarten Rijke. 2015. Media studies research in the data-driven age: How research questions evolve. *J. Am. Soc. Inf. Sci. Tec.* (2015).

[7] Dick Bulterman, Jack Jansen, Pablo Cesar, Sjoerd Mullender, Eric Hyche, et al. 2008. Synchronized Multimedia Integration Language (SMIL 3.0), W3C Recommendation, December. (2008).

[8] Paolo Ciccarese. 2013. Semantic Annotation on the Web: Domeo. (2013). https://www.youtube.com/watch?v=2VeiCwwjqcQ

[9] Paolo Ciccarese. 2014. Annotopia: Open Annotation Server. (2014). https://www.youtube.com/watch?v=2VeiCwwjqcQ&t

[10] Paolo Ciccarese, Marco Ocana, and Tim Clark. 2012. Open semantic annotation of scientific publications using DOMEO. *Journal of biomedical semantics* 3, 1 (2012), S1.

[11] Stamatia Dasiopoulou, Eirini Giannakidou, Georgios Litos, Polyxeni Malasioti, and Yiannis Kompatsiaris. 2011. A Survey of Semantic Image and Video Annotation Tools. In *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution*, David et al. Hutchison (Ed.). Vol. 6050. Springer Berlin Heidelberg, Berlin, Heidelberg, 196–239.

[12] Francesca Di Donato, Christian Morbidoni, Simone Fonda, Alessio Piccioli, Marco Grassi, and Michele Nucci. 2013. Semantic annotation with Pundit: a case study and a practical demonstration. In *Proceedings of the 1st international workshop on collaborative annotations in shared environment: metadata, vocabularies and techniques in the digital humanities.* ACM, 16.

[13] Richard Furuta and Eduardo Urbina. 2002. On the characteristics of scholarly annotations. In *Proceedings of the thirteenth ACM conference on Hypertext and hypermedia.* ACM, 78–79.

[14] B. Haslhofer, W. Jochum, R. King, C. Sadilek, and K. Schellner. 2009. The LEMO annotation framework: weaving multimedia annotations with the web. *Int. J. on Digital Libraries* 10, 1 (2009), 15–32. https://doi.org/doi.org/10.1007/s00799-009-0050-8

[15] Catherine C. Marshall. 1997. Annotation: from paper books to the digital library. In *Proceedings of the Second ACM International Conference on Digital Libraries*. New York: ACM, 131–140. https://doi.org/10.1145/263690.263806

[16] Catherine C. Marshall. 1998. Toward an ecology of hypertext annotation. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia : Links, Objects, Time and Space; structure in Hypermedia Systems (HYPERTEXT '98).* ACM, New York, NY, USA, 40–49. https://doi.org/10.1145/276627.276632

[17] Liliana Melgar, Marijn Koolen, Hugo Huurdeman, and Jaap Blom. 2017. A Process model of Scholarly Media Annotation. In *Proceedings of the 2017 ACM Conference on Human Information Interaction and Retrieval, CHIIR 2017.*

[18] Liliana Melgar Estrada and Marijn Koolen. 2017. Audiovisual media annotation using Qualitative Data Analysis Software: a comparative analysis. *The Qualitative Report* Manuscript submitted for publication (2017).

[19] Christian Morbidoni and Alessio Piccioli. 2015. Curating a Document Collection via Crowdsourcing with Pundit 2.0. In *The Semantic Web: ESWC 2015 Satellite Events*, Fabien Gandon et al. (Eds.). Vol. 9341. Springer International Publishing, Cham, 102–106. DOI: 10.1007/978-3-319-25639-9_20.

[20] Lyndon Nixon and Raphael Troncy. 2014. Survey of Semantic Media Annotation Tools for the Web: Towards New Media Applications with Linked Media. In *The Semantic Web: ESWC 2014 Satellite Events*, Valentina Presutti, Eva Blomqvist, Raphael Troncy, Harald Sack, Ioannis Papadakis, and Anna Tordai (Eds.). Vol. 8798. Springer International Publishing, Cham, 100–114.

[21] Marco Ocana, Paolo Ciccarese, Nunzio, and Timothy William Clark. 2012. Open semantic annotation of scientific publications using DOMEO. *Journal of Biomedical Semantics* 3, Suppl 1 (2012). https://doi.org/10.1186/2041-1480-3-S1-S1

[22] Carole L. Palmer, Lauren C. Teffeau, and Carrie M. Pirmann. 2009. *Scholarly information practices in the online environment: themes from the literature and implications for library service development.* Technical Report. OCLC Research.

[23] R. Sanderson, P. Ciccarese, and H. Van de Sompel. 2013. Open Annotation Data Model. W3C Community draft, 13 February 2013. http://www.openannotation.org/spec/core/. (February 2013).

[24] Robert Sanderson, Paolo Ciccarese, and Benjamin Young. 2017. Web Annotation Data Model: W3C Recommendation 23 February 2017. (2017).

[25] Pavel Smrz and Jaroslav Dytrych. 2015. Advanced Features of Collaborative Semantic Annotators: The 4A System. In *The Twenty-Eighth International Flairs Conference.*

[26] John Unsworth. 2000. Scholarly Primitives: what methods do humanities researchers have in common, and how might our tools reflect this? Humanities Computing: formal methods, experimental practice, May 13, London, UK. (2000).

[27] Niels-Oliver Walkowski. 2016. The Landscape of Digital Annotations and Its Meaning. In *Proceedings of the Conference on Language Technologies and Digital Humanities*, Tomaž Erjavec and Darja Fišer (Eds.). Academic Publishing Division of the Faculty of Arts, Ljubljana, Slovenia, 6–11.

[28] Niels-Oliver Walkowski and E. T. E. Barker. 2014. Digital humanists are motivated annotators. In *Digital Humanities 2014, Laussane, Switzerland.*

[29] Richard Wallis, Antoine Isaac, Valentine Charles, and Hugo Manguinhas. 2017. Recommendations for the application of Schema.org to aggregated Cultural Heritage metadata to increase relevance and visibility to search engines: the case of Europeana. *Code4Lib* 36 (2017).

[30] Megan Winget. 2013. A meditation on social reading and its implications for preservation. *Preservation, Digital Technology & Culture* 42, 1 (2013), 39–52. https://doi.org/10.1515/pdtc-2013-0004