Fostering Interaction with Cultural Heritage Material via Annotations: The FAST-CAT Way

Nicola Ferro[1], Gary Munnelly[2], Cormac Hampson[2], and Owen Conlan[2]

[1] Department of Information Engineering, University of Padua, Italy
ferro@dei.unipd.it

[2] The University of Dublin, Trinity College, Ireland
{munnelg, cormac.hampson, owen.conlan}@scss.tcd.ie

Abstract. This paper describes the innovative annotation facilities of the CULTURA portal for digital humaties, which are aimed at improving the interaction of non specialist users and general public with cultural heritage contents. The annotation facilities are comprised by two modules: the FAST annotation service as back-end and the CAT Web front-end integrated in the CULTURA portal.

1   Introduction

Almost everybody is familiar with annotations and has his own intuitive idea about what they are, drawn from personal experience and the habit of dealing with some kind of annotation in everyday life, which ranges from jottings for the shopping to taking notes during a lecture or even adding a commentary to a text. This intuitiveness makes annotations especially appealing for both researchers and users: the former propose annotations as an easy understandable way of performing user tasks, while the latter feel annotations to be a familiar tool for carrying out their own tasks. Therefore, annotations have been adopted in a variety of different contexts, such as content enrichment, data curation, collaborative and learning applications, and social networks, as well as in various information management systems, such as the Web (semantic and not), digital libraries, and databases.

The role of annotations in digital humanities is well known and documented [1-6]. Subsequently, many different tools which allow for the annotation of digital humanities content have been developed. Unfortunately, tools designed specifically for an individual portal are typically only compatible with that system. More general solutions, which can be easily distributed across various sites, have been developed, but these systems often have limited functionality (only annotating a single content type, no sharing features etc.) [7-8].

FAST-CAT (Flexible Annotation Semantic Tool - Content Annotation Tool) is a generic annotation system that directly addresses this challenge by providing a convenient and powerful means of annotating digital content. This paper introduces FAST, the backend service providing powerful annotation functionalities, and CAT, the frontend Web annotation tool, and discusses how its features are tackling important challenges within the Digital Humanities field.

FAST-CAT is being developed as part of the CULTURA project [9-10]. A key aspect of CULTURA is the production of an online environment that empowers users, of various levels of expertise, to investigate, comprehend and contribute to digital cultural collections. FAST-CAT is a key component of this environment and is currently being trialed with the help of three different user groups.

The paper is organized as follows: Section 2 explains the fast annotation model and the search functionalities on top of it; Section 3 describes the CAT annotation interaction model; Section 4 introduces the FAST-CAT architecture; Section 5 discusses the CULTURA environment; and, Section 6 draws some conclusions and outlook future work.

2   FAST Annotation Model

The FAST annotation service adopts and implements the formal model for annotations proposed by [3] which has been also embedded in the reference model for digital libraries developed by DELOS, the European network of excellence on digital libraries [11].

According to this model, an annotation is a compound multimedia object which is constituted by different signs of annotation. Each sign materializes part of the annotation itself; for example, we can have textual signs, which contain the textual content of the annotation, image signs, if the annotation is made up of images, and so on. In turn, each sign is characterized by one or more meanings of annotation, which specify the semantics of the sign; for example, we can have a sign whose meaning corresponds to the title field in the Dublin Core (DC) metadata schema, in the case of a metadata annotation, or we can have a sign carrying a question of the author's about a document whose meaning may be "question" or similar.

An annotation has a scope which defines its visibility (public, shared, or private), and can be shared with different groups of users. Public annotations can be read by everyone and modified only by their owner; shared annotations can be modified by their owner and accessed by the specified list of groups with the given access permissions, e.g. read only or read/write; private annotations can be read and modified only by their owner.

Figure 1 shows an example of annotation which summarizes the discussion so far. The annotation, with identifier `a1` and namespace `fast`, is authored by the user `ferro`. It annotates a document containing a novel, whose identifier is `doc1` and which belongs to the namespace `dl1` of a digital library which manages it. The annotation relates to another document containing a translation of the novel, whose identifier is `doc35` and which belongs to the namespace `dl2` of a digital library different from the one which manages `doc1`; in addition, it relates also to the Web page of the publisher of the novel, whose identifier is `http://www.publisher.com/` and which belongs to the namespace `fweb`, used for indicating Web resources.

In particular, `a1` annotates two distinct parts of `doc1`. It annotates an image contained in the PDF of the novel by using a textual sign whose content is "This is a common picture for this novel" and whose meaning is to be a `comment` in the `fast` namespace. It also annotates a sentence by using another textual sign whose content is

"Description of the picture" and whose meaning is to be a `comment` in the `fast` namespace.
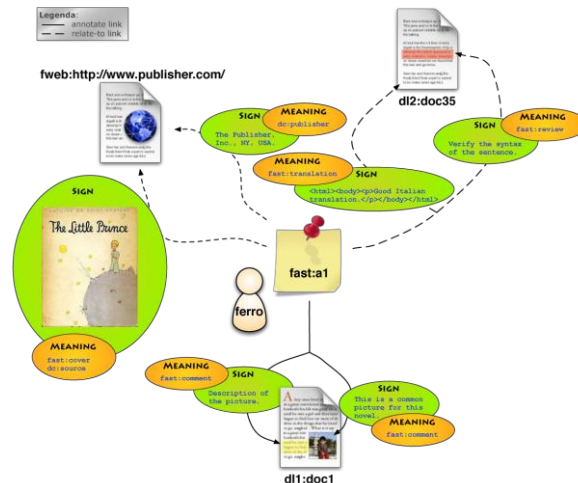

Figure 1: Example of annotation.

`a1` relates the document `doc1` to its Italian translation by linking to the whole document `doc35` with a textual sign whose content is "Good Italian translation" and whose meaning is to be a `translation` in the `fast` namespace. It also relates to a specific sentence of the translation with a HTML sign which asks to "Verify the syntax of the sentence" and whose meaning is to be a `review` in the `fast` namespace. Finally, `a1` also relates the document to the Web page of the publisher of the novel with a textual sign whose content is "The Publisher, Inc., NY, USA" and whose meaning is to be the `publisher` field of the `DC` metadata schema. It also relates the document to the Web page of the publisher via an image sign, containing the cover of the printed book of the novel by the publisher, and whose meaning is to be both a `source` field in the `DC` metadata schema and a `cover` in the `fast` namespace.

The flexibility inherent in the annotation model allows us to create a connective structure, which is superimposed to the underlying documents managed by digital libraries. This can span and cross the boundaries of different digital libraries and the Web, allowing the users to create new paths and connections among resources at a global scale.

2.1 Search Model

The presence of both structured and unstructured content within the managed resources calls for different types of search functionalities, since structured content can be dealt with exact match searches while unstructured content can be dealt with best match searches. These two different types searches may need to be merged together in a query if, for example, the user wants to retrieve annotations by a given author about a given topic; this could be expressed by a boolean AND query which specifies both the author (structured part) and the content (unstructured part) of the annotations to be

searched. Nevertheless, boolean searches are best suited for dealing with exact match searches and they need to be somewhat extended to also deal with best match searches. Therefore, we need to envision a search strategy able to express complex conditions that involve both exact and best match searches. The "P-norm" extended boolean model proposed by [12] is capable of dealing with and mixing both exact and best match queries, since it is an intermediate between the traditional boolean way of processing queries and the vector space processing model. Indeed, on the one hand, the P-norm model preserves the query structure inherent in the traditional boolean model by distinguishing among different boolean operators (and, or, not); on the other hand, it allows us to retrieve items that would not be retrieved by the traditional boolean model due to its strictness, and to rank them in decreasing order of query-document similarity. Moreover, the P-norm model is able to express queries that range from pure boolean queries to pure vector-space queries, thus offering great flexibility to the user.

The hypertext that connects documents to annotations calls for a search strategy that takes it into consideration and allows us to modify the score of annotations and/or documents according to the paths in the hypertext. For example, we could consider that an annotation, retrieved in response to a user query, is more relevant if it is part of a thread where other annotations have also been retrieved in response to the same query rather than if it is part of a thread where it is the only annotation that matches the query.

The FAST Context Set [13] has been defined in order to provide a uniform query syntax to FAST by using the Contextual Query Language (CQL) [14], developed and maintained by the Library of Congress in the context of the Z39.50 Next Generation (ZING) project. FAST provides conformance to CQL up to Level 2.

3 CAT Annotation Interaction Model

CAT is a web annotation tool developed with the goal of being able to annotate multiple types of documents and assist collaboration in the field of digital humanities. At present, CAT allows for the annotation of both text and images. The current granularity for annotation of text is at the level of the letter. For image annotations, the granularity is at the level of the pixel. This allows for extremely precise document annotation, which is very relevant to the Digital Humanities domain due to the variety of different assets that prevail. How this precision was achieved is discussed in section 3.1.

There are two types of annotation which may be created using CAT; a targeted annotation and a note. A targeted annotation is a comment which is associated with a specific part of a document. This may be a paragraph, a picture or an individual word, but the defining feature is that the text is directly associated with a specific subset of the digital resource. Conversely, a note is simply attached to the document. It is not associated with a specific item therein. Typically, this serves as a general comment or remark about the document as a whole.

In addition to allowing a user to comment on document text, the annotations created using CAT allow an individual to link their annotations to other, external sources. This is hugely beneficial for teachers using digital cultural collections and for students

from primary to university level as well as experienced researchers. As can be seen in Figure 1, the addition of links to a resource greatly enriches the amount of information it contains. Importantly, each link has comment text associated with it, allowing an educator to explain why this specific link is important or what the student should seek to gain from reading this particular source.
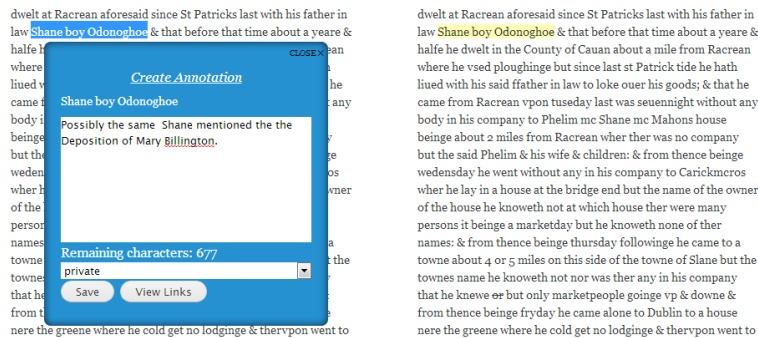

Figure 2: User creates a targeted annotation on a body of text about a person of interest

While CAT is beneficial for researchers and educators, it is also being used as an important source of user data for the content provider. Websites such as Amazon[1] and YouTube[2] are able to provide increasingly accurate recommendations for their individual users. These recommendations are facilitated by a user model which is driven by a combination of factors such as ratings and recently viewed items. For a digital humanities site, annotations can provide an insight into which entities are of interest to a user. If a user is frequently annotating a document, it is likely that this document is of interest to them. Furthermore, if the text being annotated is analysed, it may be possible to discern specific entities of interest within the document. A digital humanities site which could recommend resources that are relevant to users would be profoundly useful, and would help improve the effectiveness with which researchers interact with their domain.

3.1   Annotation Pointers

In order for the tool to provide the desired degree of annotation precision, it must have a reliable means of storing a pointer to the section that is being annotated. This is achieved by serializing the selected region into a string which can be parsed and used to reconstruct the selected range independently of the initial selection.
For text, this serialized representation takes the form:

```
<PathStart>;<OffsetStart>;<PathEnd>;<OffsetEnd>
```

Where:

- `<PathStart>` is the path to the element which contains the start of the user's selection.

---

[1] http://www.amazon.co.uk/
[2] http://www.youtube.com/

- `<OffsetStart>` is the offset into the start element where the beginning of the selected text may be found.
- `<PathEnd>` is the path to the element which contains the end of the user's selection.
- `<OffsetEnd>` is the offset into the end element where the end of the selected text may be found.

For images, the form is:

```
<Path>;<OffsetX>;<OffsetY>;<AnnotationH>;<AnnotationW>
```

Where:

- `<Path>` is the path to the annotated image.
- `<OffsetX>` and `<OffsetY>` are the position of the upper left corner of the annotation.
- `<AnnotationH>` and `<AnnotationW>` are the height and width of the annotation within the image.

In both cases, the `path` is computed using a modified version of the open source Okfn annotator [7] range class. In order to improve cross browser compatibility, CAT replaces Okfn's XPath pointers with CSS selectors. There are two reasons for this change. Firstly, different browsers will render pages in different ways, which means that XPath is not always a reliable means of locating a specific element in the markup. Secondly, support for XPath has been removed from current releases of jQuery. CSS selectors, however, are still supported and hence are the more suitable choice.

Additionally, rather than using browser ranges, CAT uses Rangy [20] ranges. Rangy is an open source JavaScript library which creates a virtual representation of a selected range that is independent of the browser being used. Rangy can then map this virtual range to the current page, taking into consideration the browser being used. Pointers are generated with respect to this virtual range so that the result should always evaluate to the same document location regardless of the environment.

FAST provides a `pointer` field as part of an annotation's representation. This is a free-text field, allowing CAT to define its own format for indicating the section of a document with which an annotation is associated. The serialized representation of the annotated range is stored at this location.

4   Architecture

4.1   FAST Architecture

The FAST annotation service comprises three sub-systems:

- **logging infrastructure**: lays behind all the components of the FAST system, captures information such as the user name, the IP address of the connecting host, the action that has been invoked by the user, the messages exchanged among the components of the system in order to carry out the requested action, any error condition, and so on. Moreover, as far as the FAST RESTful Web Application is concerned, it captures also the HTTP logs and represents them according to the W3C Extended Log File Format [15]. Furthermore, the log events can be accessed and searched interactively by means of (possibly) complex extended

Boolean queries, comprising both exact and best match clauses, giving thus the possibility to mine and fully exploit them;
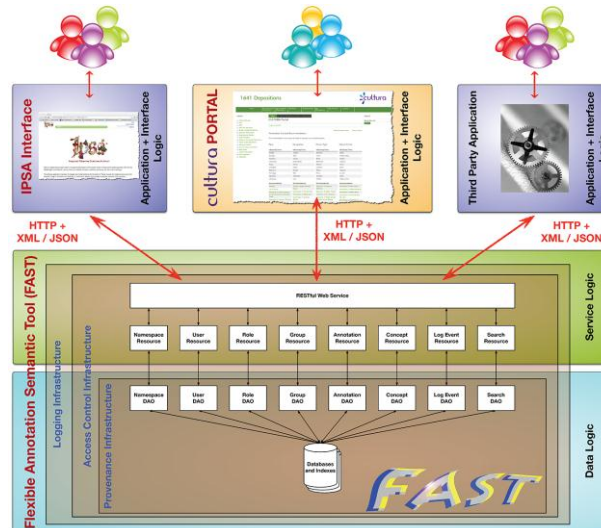

Figure 3: Architecture of the FAST annotation service.

- **access control infrastructure**: takes care of monitoring the access to the various resources and functionalities offered by the system. On the basis of the requested operation, it performs: (i) authentication, i.e. it asks for the user credentials before allowing to perform an operation; (ii) authorization, i.e. it verifies that the user currently logged in holds sufficient rights to perform the requested operation; The access control policies can be dynamically configured and changed over the time by defining roles, i.e. groups of users, entitled to perform given operations. This allows institutions to define and put in place their own rules in a flexible way according to their internal organization and working practices. Moreover, the access control infrastructure provides fine-grained control over the access to the specific resources, based on the permission granted to the resources, e.g. only the owner of a private resource and read it, even if the reading of that resource is granted to all roles;
- **provenance infrastructure**: keeps fine trace, for each resource managed by the system, of its full lineage since its first creation, allowing us to reconstruct its fully history and modifications over the time. Provenance events are statements about a resource of the form: `<when>` `<who>` `<predicate>` `<what>` `<why>` where `<when>` is the time stamp at which the event occurred; `<who>` is the user who caused the event; `<predicate>` is the action carried out in the event, i.e. CREATED, READ, or DELETED; `<what>` is the resource originated by the event, i.e. a dump of the actual content of the resource; and `<why>` is the motivation that originated the event, i.e. the operation performed by the system that led to a modification of the resource. For all these events, a dump of a resource is stored in the Provenance Infrastructure, thus allowing us to access to the

different versions of it over the time, even after it has been deleted from the system.

The FAST annotation service is exposed as a RESTful Web Service [16] which allows for the development of different applications and plug-ins over it in an open, collaborative, and scalable way which ensure sustainability over the time.
The FAST annotation service has been developed by using the Java[3] programming language, which ensures good portability of the system across different platforms. We used the PostgreSQL[4] DataBase Management System (DBMS) for the actual persistence of annotations and its full text extension for indexing and searching the full text components of the managed resources. The Apache Tomcat[5] Web container and the Restlet[6] framework have been used for developing the FAST RESTful Web Application.

4.2   CAT Architecture

The architecture of the CAT annotation tool is comprised of two layers; A client-side front end, coded using JavaScript and jQuery, and a Drupal 7 module back end, written in PHP.
The front end runs in the user's browser and provides them with a user interface through which they can interact with annotations. When a user has chosen a particular course of action, the data is passed into the logic module where their request can be processed. Depending on the nature of the request, certain third party libraries may be used in the procedure. For example, in the process of annotating a text object, the location of the text in the document must be recorded in a cross platform manner. In order to do this, a representation of the highlighted range is generated using rangy. This is a purely virtual range which means it is slightly slower than using the browser's range, but it has the advantage of being cross platform. Using a modified version of the Okfn path finder, the logic then computes a serialized path to the selected location represented by rangy which can be stored as a pointer in FAST. When annotating images, the process is the same except that jCrop [21] provides details of the selected region rather than Rangy. Retrieving an annotated region is simply the reverse of this process.
The representation of an annotation created here is a simplified version of the FAST description of the annotation. This is to minimize the amount of data that a user must send and receive to and from the server. For example, details such as namespaces are added on the back end rather than on the front end (and thus are managed by the site administrator). Furthermore, when managing details such as groups, the user's permissions are derived from the verbose annotation description on the server and then passed as a single value in the simplified representation.

---

[3] http://www.oracle.com/technetwork/java/index.html
[4] http://www.postgresql.org/
[5] http://tomcat.apache.org/
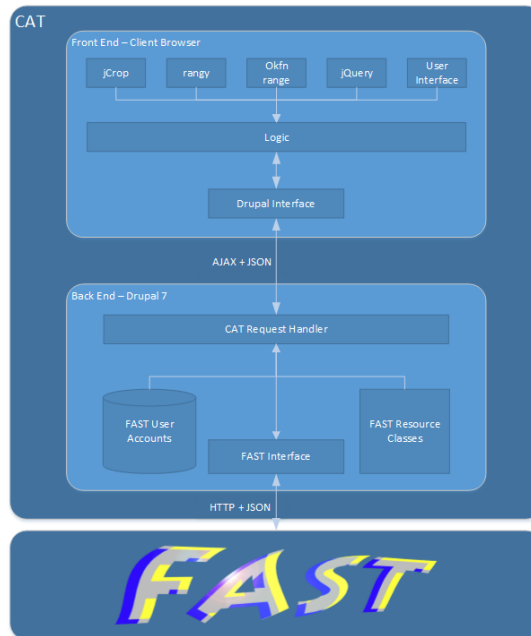[6] http://www.restlet.org/

Figure 4: Architecture of the CAT annotation tool

The Drupal 7 module on the back end acts as a relay between FAST and the user. Requests for annotation creation, deletion, download etc. are passed from the front end to a request handler function on the back end. This callback function structures the data sent by the front end so that it conforms to the FAST schema and then generates the HTTP packets to be transferred. There is some logic applied at this point to determine which packets need to be sent and in what order for the request to be fulfilled. Once the system is ready, the packets are sent on to FAST. The Drupal module then waits for a response from the remote service. When one is received, the result is returned to the front end via the same callback function through which the request was initially made.

The choice of a Drupal module as a means of implementation means that adding FAST-CAT to any site using the Drupal CMS should be a very simple process. Additionally, as the Drupal module is only acting as a relay, it should be a relatively simple process to swap out the back end for a more server agnostic implementation, allowing FAST-CAT to be deployed on any website, rather than only those using the Drupal 7 content management system.

Certain requests such as creating and viewing annotations require user authentication by FAST. As FAST is a stand-alone service, it maintains its own record of user accounts and login details. This means that for each user who is registered on the CULTURA site (see section 5), a separate account must be created for them in FAST. CAT performs this registration automatically.

5   The CULTURA Environment

CULTURA[7] is a three year, FP7 funded project, scheduled to finish in February 2014. Its main objective is to pioneer the development of personalised information retrieval and presentation, contextual adaptivity and social analysis in a digital humanities context. In its current form, it aims to provide adaptive and personalized access to two historical collections – the 1641 depositions [17] and IPSA [18].

FAST-CAT has been integrated into the environment in order to provide users with an additional means of interacting with the portal, as well as to help provide some feedback for CULTURA's user model regarding a user's interests. At present, CULTURA (and by extension FAST-CAT) is being evaluated by three groups of users.

A team of MPhil students and professional researchers from Trinity College Dublin are using FAST-CAT as part of their teaching, collaboration and research into the 1641 depositions. These users will be testing the annotation tool in a free form manner. How they choose to annotate and what content they label is entirely determined by their own needs.

The 1641 depositions are a collection of handwritten witness statements taken from Protestant men and women of all classes of society during the Catholic rebellion of 1641. These documents provide an incredible insight into the state of Ireland, Scotland and England in the period surrounding the rebellion and are an unparalleled source of information in this field. The depositions are textual in content, so these students will serve only to evaluate the text annotation aspect of the tool.

Providing an alternative insight to FAST-CAT is a group of secondary school students from Lancaster who used the annotations as part of a project they were given during a lesson. Their experience was more guided than that of the masters students as they were directed to highlight information or points of interest using FAST-CAT and then deliver a presentation using annotations to help with organization. The focus of this lesson was on the 1641 depositions.

Masters students in Padua will test the image annotation functionality of FAST-CAT as part of their research into the Imaginum Patavinae Scientiae Archivum (IPSA) [18] collections of illuminated manuscripts.

The IPSA manuscripts are a series of illustrated documents which describe the various properties of herbs and plants dating from as far back as the 14th century. They have the very rare and wonderful quality of having been incredibly accurately and realistically hand drawn from nature. While there is a Latin commentary for each plant, the real interest in these documents lies in the illustrations.

Similarly to the MPhil students, the approach of these masters students to annotating documents will be determined by their own research methodology. The intention is not to guide the users on how to use FAST-CAT, but rather to make them aware of the functionality provided and observe how they choose to apply it.

The various features offered by FAST-CAT and its user interface will be evaluated in detail and comparisons will be drawn between the manner in which different user groups availed of annotations depending on their level of expertise and the type of documents examined. Furthermore, FAST-CAT will also help to drive CULTURA's

---

[7] http://www.cultura-strep.eu/

comprehensive user model by providing the site with updates on the user's behaviour regarding document annotation.

6   Conclusions

It is the belief of the authors that FAST-CAT has huge potential as an annotation tool within the digital humanities field. However, it is still a young tool with much room for future expansion and enhancement. Some of the required additions are already known and are currently being developed. Others will be dependent on user feedback from test groups as they identify issues the experienced within their domains.
A large facet of plans to improve FAST-CAT is to increase the range of content types with which it may be used. At present, it provides for the annotation of text and images. Possible additions to this list include dynamic content types such as SVDs.
As was mentioned in section 4.2, it is possible to make FAST-CAT more server agnostic by swapping out the Drupal 7 back end for a more general php script. It is expected that this script will be developed and provided with future versions of FAST-CAT so as to increase the range of portals to which it may be applied
Further to this, another part of the future development of FAST-CAT will be focused on improving the user's experience. It is intended that the tool be as intuitive and easy to use as possible. How this will be achieved is to be this based on the feedback given by the user groups during the CULTURA trials.

Acknowledgements

References

1. Agosti, M., Ferro, N., Frommholz, I., and Thiel, U. (2004). Annotations in Digital Libraries and Collaboratories - Facets, Models and Usage. In Heery, R. and Lyon, L., editors, *Proc. 8th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2004)*, pages 244-255. Lecture Notes in Computer Science (LNCS) 3232, Springer, Heidelberg, Germany.
2. Agosti, M., Bonfiglio-Dosio, G., and Ferro, N. (2007). A Historical and Contemporary Study on Annotations to Derive Key Features for Systems Design. *International Journal on Digital Libraries*, 8(1):1-19.
3. Agosti, M. and Ferro, N. (2008). A Formal Model of Annotations of Digital Content. ACM Transactions on Information Systems (TOIS), 26(1):3:1-3:57.
4. Bélanger, M.-E. (2010, 02 03). Ideals. Retrieved 10 25, 2012, from https://www.ideals.illinois.edu/bitstream/handle/2142/15035/belanger.pdf?sequence=2

---

[8] http://www.promise-noe.eu/

5. Barbera, N., Meschini, F., Morbidoni, C., and Tomasi, F.. (2012). Annotating digital libraries and electronic editions in a collaborative and semantic perspective. In Agosti, M., Esposito, F., Ferilli, S., and Ferro, N., editors, Taking Care of Legacy and Cultural Heritage Documents - Proc. 8th Italian Research Conference (IRCDL 2012), pages 46-57. Communications in Computer and Information Science (CCIS) 354, Springer, Heidelberg, Germany.

6. Ferro, N. and Silvello, G. (2012). Empowering Archives Through Annotations. In Agosti, M., Esposito, F., Ferilli, S., and Ferro, N., editors, *Taking Care of Legacy and Cultural Heritage Documents - Proc. 8th Italian Research Conference (IRCDL 2012)*, pages 58-69. Communications in Computer and Information Science (CCIS) 354, Springer, Heidelberg, Germany.

7. Okfn. (n.d.). Okfn Annotator. Retrieved 06 2012, from http://okfnlabs.org/annotator/

8. TILE. (2011). TILE: text-image linking environment. Retrieved 07 2012, from http://mith.umd.edu/tile/

9. Hampson, C., Agosti, M., Orio, N., Bailey, E., Lawless, S., Conlan, O., and Wade, V. (2012). The CULTURA Project: Supporting Next Generation Interaction with Digital Cultural Heritage Collections. In *Progress in Cultural Heritage Preservation - 4th International Conference (EuroMed 2012)*, pages 668-675. Lecture Notes in Computer Science (LNCS) 7616, Springer, Heidelberg, Germany.

10. Hampson, C., Lawless, S., Bailey, E., Yogev, S., Zwerdling, N., Carmel, D., Conlan, O., O'Connor, A. and Wade, V. (2012). CULTURA: A Metadata-Rich Environment to Support the Enhanced Interrogation of Cultural Collections. In *Metadata and Semantics Research - 6th Research Conference (MTSR 2012)*, pages 227-238. Communications in Computer and Information Science (CCIS) 343, Springer, Heidelberg, Germany.

11. Candela, L., Castelli, D., Ferro, N., Koutrika, G., Meghini, C., Pagano, P., Ross, S., Soergel, D., Agosti, M., Dobreva, M., Katifori, V., and Schuldt, H. (2007). The DELOS Digital Library Reference Model. Foundations for Digital Libraries. ISTI-CNR at Gruppo ALI, Pisa, Italy. http://www.delos.info/files/pdf/ReferenceModel/DELOS_DLReferenceModel_0.98.pdf.

12. Salton, G., Fox, E. A., andWu, H. (1983). Extended Boolean Information Retrieval. *Communications of the ACM (CACM)*, 26(11):1022–1036.

13. Ferro, N. (2009). Annotation Search: The FAST Way. In Agosti, M., Borbinha, J., Kapidakis, S., Papatheodorou, C., and Tsakonas, G., editors, *Proc. 13th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2009)*, pages 15-26. Lecture Notes in Computer Science (LNCS) 5714, Springer, Heidelberg, Germany.

14. OASIS Search Web Services Technical Committee (2012). searchRetrieve: Part 5. CQL: The Contextual Query Language Version 1.0. http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/searchRetrieve-v1.0-part5-cql.pdf.

15. Hallam-Baker, P. M. and Behlendorf, B. (1996). Extended Log File Format – W3C Working DraftWD-logfile-960323. http://www.w3.org/TR/WD-logfile.html.

16. Fielding, R. T. and Taylor, R. N. (2002). Principled Design of the Modern Web Architecture. ACM Transactions on Internet Technology (TOIT), 2(2):115–150.

17. Trinity College Dublin, 1641 Depositions. Retrieved 11 2012, from http://1641.tcd.ie/index.php

18. Universita Degli Studi Di Padova, IPSA – Imaginum Patavinae Scientiae Archivum. Retrieved 11 2012, from http://www.ipsa-project.org/

19. Rangy, Retrieved 11 2012 from http://code.google.com/p/rangy/

20. Deep Liquid, jCrop, Retrieved 11 2012 from http://deepliquid.com/content/Jcrop.html