Strategy Evaluation for High Quality Crowd Annotations in Cultural Heritage

Master's Thesis

Bas van Sambeek

Strategy Evaluation for High Quality Crowd Annotations in Cultural Heritage

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE
TRACK INFORMATION ARCHITECTURE

by

Bas van Sambeek
born in Breda

Web Information Systems
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
http://wis.ewi.tudelft.nl

Strategy Evaluation for High Quality Crowd Annotations in Cultural Heritage

Author: Bas van Sambeek
Student id: 1308289
Email: me@basvansambeek.com

Abstract

Digitised Cultural Heritage Objects like pictures, video and audio are hard to retrieve without metadata like annotations describing what is depicted or the events which take place. Unfortunately the creation of this metadata is quite a slow and lengthy process, especially when this needs to be done in a qualitative manner. So the question arises how we can help to improve this process by increasing the speed and quantity of these annotations without sacrificing the quality. The chosen approach is to use lay people and ask them to help us with annotating these objects. As a means to guard our quality demands we want to use a taxonomy which allows the lay people to annotate in a structured manner. In order to help lay people we've created a workbench which allows us to easily create different annotation strategies, which result in different ways a taxonomy is presented to a lay person. By presenting different strategies we can investigate which strategy is more likely to help a group of lay persons when this is executed on a larger scale.

Thesis Committee:

Chair:	Prof. dr. ir. G.J.P.M. Houben, Faculty EEMCS, TUDelft
University supervisor:	Ir. J.E.G. Oosterman, Faculty EEMCS, TUDelft
Committee Member:	Dr. ir. A. Bozzon, Faculty EEMCS, TUDelft
Committee Member:	Ir. H.J.A.M. Geers, Faculty EEMCS, TUDelft

Preface

This thesis documents my journey through the worlds of crowdsourcing, Cultural Heritage and annotations. Along the way I've learned a lot about these topics, and more importantly; myself, for which I'm ever grateful. Now the time has come to say goodbye to my time at Delft University of Technology, the place where I have made many new friends, and start a new chapter in life.

While the title page may contain only one author, this thesis could not have come to pass without the help of others. First and foremost I would like to thank Jasper Oosterman, for always having an open door, guiding me back on track whenever I was lost and, most importantly, learning me how to speak Dutch properly *wink*. Next I want to thank Geert-Jan Houben for providing valuable feedback and helping me find a project in the WIS group. My gratitude also goes to Alessandro Bozzon and Hans Geers for being part of my thesis committee.

The making of this thesis would certainly have been less fun without all the coffee/tea breaks with all the, past and current, inhabitants of the master students lab on the 8$^{\text{th}}$ floor. I would like to thank them for making my time in the lab enjoyable and providing me with many (and often too much) distractions. To conclude this preface I would like to thank my parents, sister, family and friends for supporting me during my university career.

Bas van Sambeek
Delft, the Netherlands
19th December 2013

Contents

List of Figures

Chapter 1

Introduction

Assume you have been on holiday to New York with your friends and you all came back with SD cards filled with thousands of photos and hundreds of video's. During your stay you have taken some photos with yourself next to the Statue of Liberty and want to show them to your relatives, but where are they? You know they should be somewhere in the middle and you start browsing through your photo's in chronological order. As a result you have ended up with browsing through the majority of your photos in order to find the few you were looking for.

A different problem exists with the images you've received from your friends, since you didn't make those photo's you don't exactly know what's on them. Instead of browsing through all of them, wouldn't it be convenient if there was a better way to get an idea what's on them?

The same problems exists for large institutes which preserve a lot of multimedia content like the Rijksmuseum Amsterdam, Erfgoed Delft and Beeld & Geluid with the main difference that instead of dealing with thousands of items their collections can range up to a million.

1.1 Background and Motivation

While it is (relatively) easy to do a search for textual documents about a certain topic, for images this isn't the case. Unfortunately for a computer it is still hard to determine that a bag of pixels, coloured dots, actually portrays the Statue of Liberty while for a document it can simply determine if the phrase *"Statue of Liberty"* occurs within the text. Therefore it is harder to ask a computer questions like *"Show me all the images which contain the Statue of Liberty"* Although efforts are being made in the field of image recognition[1] it still isn't perfect. In the case of the Rijksmuseum (RMA) (our running example during this thesis - see Chapter 4) it is even harder to use these techniques due to the nature of the collection, which consists of hand drawn prints that contain artistic freedoms and symbolic representations. Even if a system manages to detect that a print portrays, for example, a flower; chances are low that it could determine the type of flower.

[1] Google+'s Best New Unadvertised Feature: Photo Search With Visual Recognition - Try It On Your Own Pictures And Be Amazed - http://andp.lc/ZbyGRZ

That is why various institutes have decided early on that these objects need an accompanying text attached to it, so called annotations, which can be easily indexed and retrieved. It is already fairly common that users can search images based on annotations like the title and description. However these annotations don't completely cover all the elements and events depicted on the image. That is why short additional annotations are added to these images which do just that. These annotations are similar to an old fashioned label attached to an object, for example to describe what's inside a box. In our holiday picture case the labels could for example say *"Statue of Liberty"*, *"USA"*, *"New York"*. The need to label objects has even become more urgent since many institutes are currently in the process of trying to make their collections available online for all the public to see[2]. With the use of these labels it is easier to find or recommend specific artworks in a big collection.

Although computers are better at computational tasks than humans, with regards to adding labels we still have an advantage. Humans are still the best at interpreting what they see and making connections with what they see and what they know. Of course, not every person knows everything about everything, but when given a large enough group of people, each with their specific area of expertise, we can combine that knowledge into a very precise description of an object. Take for example an image of a castle surrounded by trees. A medieval castle expert knows everything there is to know about the castle, but he will likely know very little about the trees surrounding it. When showing the same image to a botanist, he is more likely to tell you exactly which kinds of trees surround the castle then information about the castle itself.

Unfortunately, when you deal with large collections, it takes a while before each object is labelled. For example, the RMA is currently annotating 700.000 prints from their print room collection using around six professionals every day. Each professional tries to annotate about 25 images per day and as a result thus far they have roughly completed an estimated 10-15% of the collection. This means that it should roughly take another decade till the entire collection has been annotated. Due to these time constraints, the professional annotators limit themselves to a small amount of annotations per image or as they say[1]:

"We're adding 40.00 items to the collection every year. After the scan, we have limited time for each painting and this occasionally results in incomplete annotations"

However if they would try to be more complete, this would take more time they don't have, which would result in a delay of the project.

This thesis is part of the larger SEALINCMedia[3] project which has the objective to enrich Cultural Heritage (CH) collections using collaborative content curation and improve accessibility through personalised recommendation and search functionalities. Within the project various challenges are studied, one of which is the scientific understanding of the process to produce Crowd Generated Knowledge to enrich data

[2] Masterworks for One and All - http://www.nytimes.com/2013/05/29/arts/design/museums-mull-public-use-of-online-art-images.html

[3] Socially-Enriched Access to Linked Cultural Media - http://sealincmedia.wordpress.com/

collections. Producing Crowd Generated Knowledge requires understanding of three components; Goal specification, crowd identification and Activity planning [4]. Activity planning is the creation, routing and execution of tasks and the main focus of this thesis, with a main emphasis on creation and execution.

1.2 Research objectives

We want to support professional annotators with their work, by improving the speedup of the annotation process or helping them creating a broader disclosure of the objects they are labelling. We have seen other use-cases where people from outside of the organisation ('crowd annotators') can do a sufficient with aiding professionals. However we need to make sure that this doesn't affect the quality. As Oomen et al. [2] said:

"GLAMs[5] *earned their reputation over the years by preserving the quality and truthfulness of the information they offered by having full control over the acquisition, organisation and the annotation of the collection items".*

Because we don't know what would be the best method to aid these crowd annotators with making qualitative annotations, we want to create a platform where curators (the maintainers of the collections, who do not annotate) are able to create and compare different strategies. Therefore we ask ourselves the following questions:

RQ1 What are the annotation processes of crowd annotators for multimedia data?

RQ2 What are the annotation processes of Cultural Heritage professionals for digitised collections?

RQ3 Which steps are needed to allow crowd annotators to annotate on a professional level?

RQ4 How can we support curators with rapid creation and evaluation of different strategies for these steps?

1.3 Approach

In order to answer these research questions we start with a thorough analysis of what annotations are, how they are used and how we can judge their quality. After that we look at different crowdsourcing initiatives and see how their users are currently annotating objects (**RQ1**). We analyse the Print Room Online project at the Rijksmuseum and determine their current process (**RQ2**). After we have learned that a major part of ensuring the quality at the RMA comes from their usage of a taxonomy we try to find a process that combines the best of both worlds and tries to make a taxonomy accessible for lay people (**RQ3**). Based on this we create a set of requirements for an online application that allows CH institutes to create and evaluate different processes for crowd annotators (**RQ4**).

[4] The WUDE (Web User Demand Elicitation) project (WP5 of SEALINCMedia) http://www.wude.nl

[5] Galleries, Libraries, Archives and Museums

1.4 Contributions

We show that we are able to adapt the taxonomy oriented annotation approach of the professionals for the crowd. Because we don't know which method would yield the best results we have created a expandable workbench which can provide different methods of comparing and evaluating strategies. As a demo of its capabilities we have shown for one use case that the workbench is able to do so.

1.5 Outline

This thesis is structured in the following way; Chapter 2 provides a background of what annotations are and gives a quick introduction into crowdsourcing and types of metadata. Next in chapter 3 we give an overview of the current state of art with regards to annotating and in chapter 4 we do so for the Rijksmuseum. In chapter 5 we describe the desired situation where the crowd aids the RMA. We do so by creating a process which could potentially aid the external annotators. Chapter 6 describes the software we created in order to gain insight in the actions of the users. Next, in chapter 7 we evaluate if the capabilities of the workbench are sufficient to do so. Finally, in chapter 8 we present our conclusions by answering the research questions and give pointers for future work.

Chapter 2

Background

In this chapter we will give an overview of the topics which are used in this thesis.

2.1 Definitions

When reading about adding textual information to objects, one usually finds that this is usually defined by the following four words: *annotations, meta-data, tags* or *terms*. The Oxford English Dictionary defines these as follow:

Definition 1 (Meta-data[1])**.** *A set of data that describes and gives information about other data.*

Definition 2 (Tag[2])**.** *Computing To label (an item of data) in order to identify it for subsequent processing or retrieval.*

Definition 3 (Term[3])**.** *To give a particular or specified name to; to name, call, denominate, designate.*

The definition used by OED for *Annotation*[4] is quite general and not really fit for our purpose: *A note added to anything written, by way of explanation or comment*. It only talks about written objects, which is untrue in our case. Ontotext, the company behind OWLIM, a popular semantic repository for the Open Sesame RDF Framework uses the following definition[5], which is more tailored to our domain:

Definition 4 (Annotation)**.** *Annotation, or tagging, is about attaching names, attributes, comments, descriptions, etc. to a document (which can be either text or objects like images, red.) or to a selected part in a text. It provides additional information (meta-data) about an existing piece of data.*

As you can see, tagging and annotating are closely resembled. Both represent the act of adding a piece of information to an object. In this thesis we will therefore treat them

[1] meta-, prefix. meta-data Oxford English Dictionary - http://www.oed.com/view/Entry/117150

[2] tag, v.1 Oxford English Dictionary - http://www.oed.com/view/Entry/197013

[3] term, v Oxford English Dictionary - http://www.oed.com/view/Entry/199410

[4] annotation, n Oxford English Dictionary - http://www.oed.com/view/Entry/7922

[5] Semantic Annotation | Ontotext - http://www.ontotext.com/kim/semantic-annotation

as the same and use both words interchangeably. We will try to favour annotating for consistency except when referring to literature which originally refer to it as tagging.

When used in a more Computer Science setting an easier description for a term would be: *A word or a phrase that represents a concept*. A concept can be an object or something less specific as a thought or an idea. In 2.1 an overview is given how the definitions relate to each other: when we have a collection of tags, comprised of different terms, this forms a set of meta-data. This meta-data is attached to and gives information about an object.

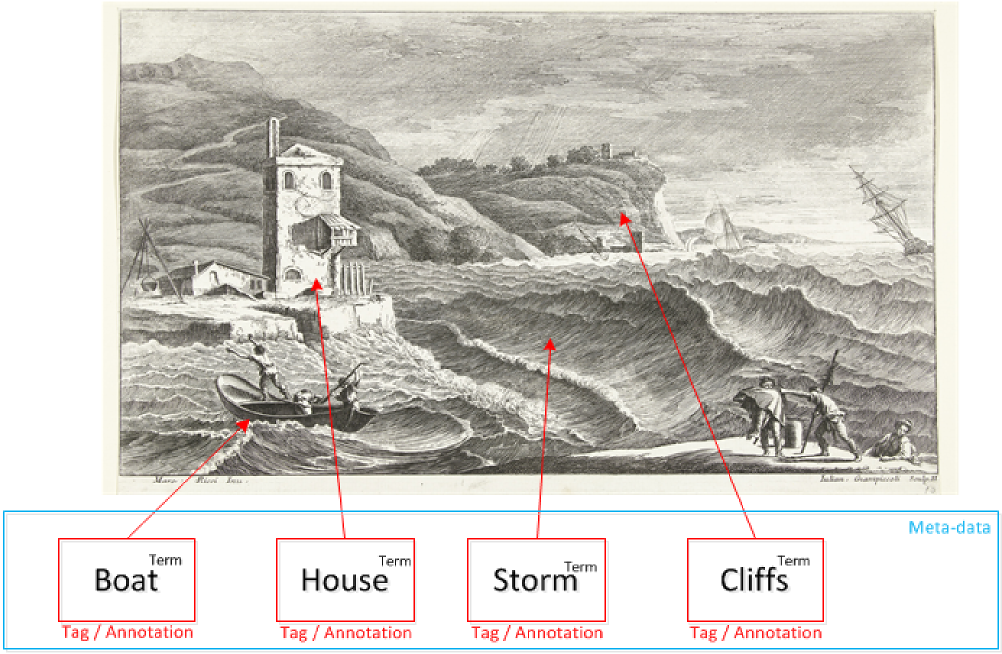


Figure 2.1: Overview how the different definitions relate to each other.

In chapter 3 we will take a closer look at annotations, their different usages, ways to add them to an object and how they are currently being used in a Cultural Heritage setting.

2.2 Crowdsourcing

Crowdsourcing *represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call* [3][4]. In this thesis these tasks are defined as activities which are relatively easy for a human to perform, but which are still complex for a computer; like recognising objects on an image or translating textual documents. Therefore this process is often called *human computation*. The group of humans can be comprised of all kinds of people, ranging from people with hardly any education to professionals in a specific field. Their motivations can be divided into different categories[5][6] where the key separation of categories can be put as monetary vs. non-monetary[7]. Non-monetary, where people perform tasks for fun, recognition or a wish to share their knowledge or monetary, in which people perform tasks in exchange for a small amount of money. Monetary tasks are usually posted on human computation marketplaces like Amazon Mechanical Turk (AMT), where tasks can be posted and each task is being rewarded with a small amount of money.

If we want to translate a book; instead of giving one entire book to one person, it is better to divide the task in hundreds of sub-tasks where many users only need to translate a paragraph. This offers a couple of advantages: Where one person will need a couple of days, maybe even weeks, to translate a complete book, when performing this task in parallel using hundreds of people the entire book can be translated in mere hours. Even though we use more people, because they are non-professionals, it will also be likely cheaper than hiring a full time translator for this task.

Another real world example is Galaxy Zoo[6]. As we all know, the universe is huge, if not infinite, and consists of over a million galaxies. Telescopes take pictures of each galaxy and for each galaxy the type of galaxy needs to be determined and, if it's an uncommon one, expensive telescope time needs to be allocated to perform a deeper investigation. Instead of using a small team of professionals, they decided to use crowdsourcing. They assumed that even with the use of crowdsourcing their initial dataset of one million galaxies would take years to complete, but within 24 hours after the launch they were receiving almost 70,000 classifications an hour. During the project's first year they received 50 million classifications, contributed by more than 150,000 persons.

But when working with a lot of different humans, with different personalities and ideas of how something should be translated, discrepancies between styles or phrasing may occur. When looking at Galaxy Zoo a lot of people can describe a galaxy in a different way. Therefore for each task multiple judgements are gathered so it can be assessed how reliable the results are. The more people say it's a star shaped galaxy, the likelier it is to be so. Another method of gaining a higher confidence in the task performed by a crowd annotator is by using gold data[8]. Gold data is a set of questions where you already know the answer to. Then when a crowd annotator performs his task he will get a mix of questions where the answers are unknown and gold questions where they are known. Only when the gold questions are properly answered in a comparable way to the correct answer the unknown answers will be trusted.

2.3 Different types of meta-data

When adding meta-data to an object you are able to choose from different vocabulary systems, with various degrees of freedom. The two important aspects we look at here are the level of difficulty for adding and removing terms and the way these terms are structured or related to each other. This section tries to give an overview of these systems in order to give a frame of mind of where this thesis is placed [9][10].

Uncontrolled vocabularies (Folksonomies)

Uncontrolled vocabularies are lists of words. These words are typically not listed in a specific order although sometimes they are displayed in alphabetic order for viewing convenience. If they were however displayed in a specific order, that would mean

[6] http://www.galaxyzoo.org/

that there is meta-data about the relationships between words and by doing so we are already moving towards a taxonomy (more about those later). As the name already suggests, uncontrolled vocabularies fall under no form of supervision; people are free to add or remove terms by their liking. Because the people are in control and they provide a self-regulating structure, they are often called folksonomies[11]. An example of the self-regulated structure is that a group of people can make agreements on which words they use for certain objects. Due to the low barrier of adding terms these folksonomies are highly adaptable and therefore can quickly be applied to new concepts. They are also cost effective, because they are controlled by a large group of people. Because there is a large group of people contributing to a folksonomy, it provides different perspectives for the same object [9].

There are disadvantages however, as over time inconsistencies or redundancies might be introduced. This can be contributed to the fact that uncontrolled vocabularies allow the use of singular and plural forms and that misspellings, differences in punctuation and capitalisation, all end up in different terms. Due to the lack of control it is also easy to add unwanted terms. This means that users can annotate your companies' content with negative terms whereas this would be impossible in a controlled setting. A popular form of using with uncontrolled vocabularies is tagging, as can be seen on social sites like Flickr, Last.fm or del.icio.us.

Controlled vocabularies

Controlled vocabularies are, in similar fashion to uncontrolled vocabularies, controlled lists of words. The distinction here is that controlled vocabulary systems are usually produced by a governing body where strict rules apply for adding or removing terms to the system, whereas in an uncontrolled setting the users have more power. Controlled vocabularies are therefore more expensive and less flexible but offer a higher quality.

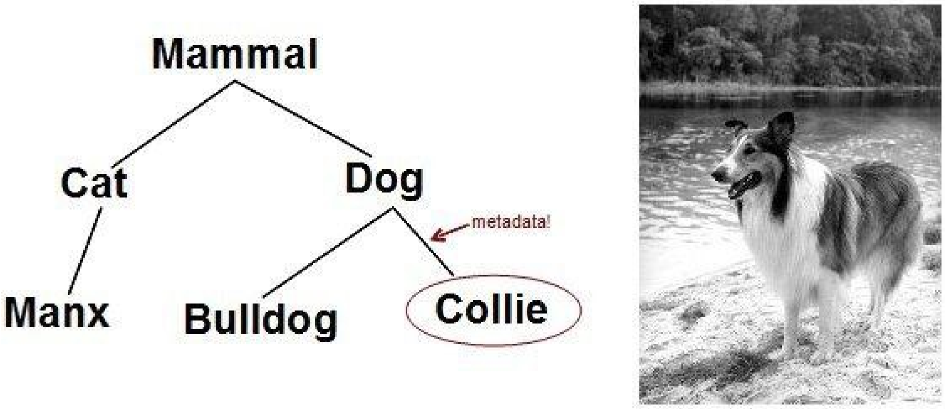


Figure 2.2: Relationships in a taxonomy. From [10].

Taxonomies

Taxonomies are a special breed of controlled vocabularies. In this case the controlled vocabulary terms are organised in a hierarchy. A taxonomy represents these relationships in broader and narrower terms. Image 2.2 shows that the term Dog is a broader term for Collie and Bulldog. Because Mammal is a broader term for Dog, Dog is a narrower term for Mammal. The broader terms can be seen as describing something more global and narrower relationships as more restrictive and precise. These relationships themselves are metadata about the terms. Using a taxonomy has the advantage that it adds value to a data collection. Take for example an image which only has been annotated with the term "collie". Then when someone searches for images of dogs, the image of the collie will be retrieved even though it not has been explicitly annotated as a dog. Using the structure in the taxonomy we know that a picture of a collie is also a picture of a dog. Besides broader and narrower relationships taxonomies can also contain alternative terms such as synonyms (for example hound as alternative for dog). In a similar fashion it is also easy to add translations of terms in other languages.

Thesauri

A thesaurus adds even more meta-data to a vocabulary then a taxonomy. Besides broader and narrower relationships, it can connect to *related* terms in a different (part of the) vocabulary. The term dog may now point to doghouse in another vocabulary to indicate that a dog uses a doghouse; as seen in figure 2.3.

Unfortunately in practice the terms taxonomy and thesaurus often interchanged, because you could argue that a taxonomy is a subset of a thesaurus. Since a thesaurus supports more relations than the broader and narrower relations, we would like to keep

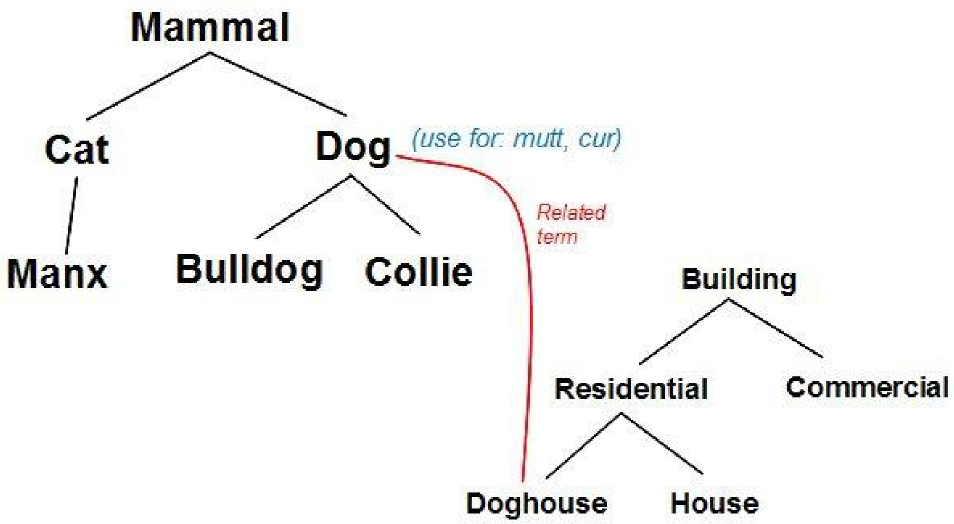


Figure 2.3: Relationships in a thesaurus. From [10].

this distinction and therefore we use the definitions of Taxonomies and Thesauri as stated above for the rest of this thesis.

Ontologies

Ontologies are not restricted to hierarchical relations between categories but allow for more elaborate and complex relations. Where in figure 2.3 we saw that the term "doghouse" was related to "dog", in an ontology we can specify that a doghouse is made of "wood", has a "roof" and in which colours they occur. As a result of that one can specify their own (business specific) relations. However, an ontology can also be more restrictive as a thesaurus. For example, the Art and Architecture Thesaurus (AAT)[7] uses two terms to represent the concept landscape: *landscape (representation)* and *landscape (environment)*[12], whereas in an ontology there should be one landscape concept. Ontologies are defined using ontology languages. The Web Ontology Language (OWL) is the W3C standard for defining ontologies.

Folksontology

The last type of meta-data is a hybrid format of two types we have seen before. A folksontology [13] tries to combine a folksonomy and a ontology and use the best of both worlds. As we have seen before a key strength of a folksonomy is that is flexible to use as users can quickly invent and add new terms. A ontology however is more rigid but thanks to it more structured approach of organising terms it enhances browsability [14] and introduces the ability to reason. A folksontology wants to categorise the generated tags so that it is easily browsable, to maintain and expand.

[7] http://www.getty.edu/research/tools/vocabularies/aat/

Chapter 3

Annotations

In this chapter we will look into annotations and the different purposes they are being used for. First we look at how annotations are being used in a Cultural Heritage setting (3.1). Next we will look at different types and purpose of annotations in section 3.2. Then we investigate and evaluate the most popular annotation initiatives which are currently being used and an Annotation Process is derived (3.3). To close this chapter we will investigate annotation quality and how the selected current initiatives perform with regards to this quality (3.4).

3.1 Annotations in a CH setting

Cultural Heritage institutions make use of different types of annotations depending on their needs. Besides describing their artworks they also need to catalogue them for various legal reasons. An example of a legal reason is knowing what your collection consists of for insurance reasons or knowing where your objects are at all time. The Getty Research Institute has compiled a list of "Categories for the Description of Works of Art" (CDWA)[15] which is *'a framework for describing and accessing information about works of art'*. Not all categories are mandatory; some categories are marked as core which is the minimum information that is required to uniquely and unambiguously identify and describe a particular work of art. Getty also makes a distinction between information intended for display versus information intended for retrieval. Information intended for display should be in a format which is easy to read and understandable by users like a title, or a textual description of an image, and therefore is usually in a free-text format. Information for retrieval however must be formatted to allow retrieval, which should be done by professionals who understand the retrieval implications for their indexing terms. A wrong approach therefore would be to parse free-text fields intended for display into retrieval indexes. CDWA therefore advises that retrieval fields should be filled with controlled vocabularies.

When looking at the Getty Metadata Standards Crosswalk [1], which provides an overview of different standards for describing works of art it can be seen that their CDWA can be mapped to a number of other standardisation initiatives. Because according to the Crosswalk the CDWA is similar or sometimes even more elaborate then other initiatives we deem it fit as a sufficient source for our analysis.

[1] http://www.getty.edu/research/publications/electronic_publications/intrometadata/crosswalks.html

#	Element Category	Suitable for Externals
5	Styles/Periods/Groups/Movements	Maybe
16	Subject matter Core	Yes
18	Descriptive note	Maybe

Table 3.1: CDWA Categories suitable for an external. For full explanations of these descriptions please see [15]. For the complete analysis see Appendix A.

Because we want to let crowd annotators help the CH institute we need to know which tasks they could complete. In order to do so we have evaluated for each category a CH professional needs to complete whether or not a crowd annotator is capable of completing it. The full analysis of all thirty-two categories can be found in Appendix A and table 3.1 lists the categories we deemed fit for externals. The criteria we based this evaluation on is that a crowd annotator should be able to create this information in a qualitative way, with as only source of information a digital representation of the object, without having a formal art degree. Another constraint is that a lot of information already needs to be entered by a professional when they first register the object in their organisation. For example, *Titles or Names* has been deemed unfit because objects always have a single name and therefore it doesn't make sense for externals to let them create one. *Materials and techniques* is deemed too difficult because it would be hard to make a proper distinction on used techniques when all the external party gets to see is an image representing the object. Therefore the only three suitable categories are *Styles/Periods/Groups/Movements*, *Descriptive Note* and *Subject matter*. However the first two are listed as a maybe: A *Descriptive Note* is a textual description of the object and although externals are certainly fit to write down a description in free-text form of what they see, it is dubious on how useful it would be: what should we do with all those descriptions? When we get hundreds of descriptions, how should we combine those in one ultimate description that will end up next to the picture on the wall of the museum? *Styles/Periods/Groups/Movements* is a maybe because although it is likely that the majority of people doesn't know all the different art styles or movements there are likely people out there who do know the differences between styles. There are also opportunities to teach them, by showing example images and asking in which resembles the annotatable print the most like they do at Galaxy Zoo, although this would make it a categorisation task instead of a description one.
The only category that we gave a definitive 'yes' for being suitable for an external is the annotation of *Subject Matter*. Getty's definition of annotating *Subject Matter* is:

Definition 5 (Subject Matter)**.** *"The subject matter of a work of art (sometimes referred to as its content) is the narrative, iconographic, or non-objective meaning conveyed by an abstract or figurative composition. It is what is depicted in and by a work of art. It also covers the function of an object or architecture that otherwise has no narrative content."*

This is a task externals should be able to do because all the information they need is contained within the digital representation of the image. Therefore due to the previous concerns we deem only the task of annotating *Subject Matter* fit for externals, which will be our focus for the rest of this thesis.

Op. type	Description	Input	Output
Choice	The annotator selects up to n items	Max n of items that can be selected	At most n selected items
Like	The annotator adds like (unlike) annotations to some items		Number of likes for each item
Score	The annotator assigns a score (from 1..n) to some items	Max value for the score	Average score of each item
Tag	The annotator describes the object with different terms	Max number of tags that can be associated	Set of tags for each item
Categorisation	Assign an object to one or more categories	Set of predefined categories	Set of categories for each item
Order	The annotator reorders the (top n) items in the input list	Number of top elements to be ordered	Position of each item
Insert/Delete	The annotator inserts/deletes up to n items in a list	Max number of items that can be inserted/deleted	Inserted objects / Tagging of deleted objects
Modify	The annotator changes the values of attributes of some items in the list	Set of modifiable attributes	Set of changes to modifiable attributes
Group	The annotator clusters the items in to n distinct groups	Max number of groups	Assignment of each item to a group

Table 3.2: List of operation types from Bozzon et al. [7]

3.2 Different types of annotations

Now we know which task a crowd annotator should perform, we would like to take a closer look at the different types of annotations they're able to annotate an object with. Annotations are available in different formats and can be added to objects in different ways. Users can be asked explicitly for an annotation, but annotations can also be gathered implicitly based on the actions of a user. Therefore in this section we will describe which different flavours of annotations are available, how they are added and the different purposes they serve.

3.2.1 Adding annotations

When we discussed annotations in the previous chapter we mainly used one particular type; where an user adds a term to an object in order to describe it (Tag). But this is not always the case, for example, adding a like to a photo on Facebook is also an annotation (Like) and the same goes for ordering a list of music artists into a top five (Order): both give an implicit annotation of how much a user appreciates the object(s). When we look at different ways annotations can be added Bozzon et al.[7] have aggregated a good list of operation types as can be seen in table 3.2. The table lists different actions, or operations, of how an annotation is added to an object. When looking at the table we can make a distinction of operation types where a user makes a selection of a list of predefined items (Choice, Categorisation, Order or Group), assign a numeric value to an item (Like, Score) or is able to modify or add annotations they came up with themselves (Tag, Insert/Delete, Modify). Although it might not seem obvious at all times that a user makes a selection or adds a numeric value, when looking at a higher level that is what happens in the background. By ranking one item higher than another, when using Order, the user indicates that he thinks the highest ranking item is a better fit. Even when using a Like an implicit binary score is given.

In order to get an idea of how complex each task is for an annotator we have made

Annotation type	Level of freedom
Tag	High
Insert, Delete	Medium
Modify	Medium
Categorisation	Low
Group	Low
Choice	Low
Like	Low
Score	Low
Order	Low

Table 3.3: Annotation types adapted from Bozzon et al. [7]

an assessment of the level of freedom a user has. This assessment has been made using a 3-point Likert Scale [16] as this should provide us with a sufficient indication of the levels of freedom. This assessment can be found in table 3.3. The idea behind this is that the more options a user has when making annotations, the harder it gets to add the correct or optimal annotation. For example, when a user adds a `Tag` to an object he is allowed to use every word in the dictionary. When looking at the Oxford English Dictionary this means that a user has more then 231,100[2] options for deciding which word to use. Of course this number increases dramatically when we add plural forms and start to construct sentences in which case the number of options goes in to the millions. Therefore `Tag` has received a high level of freedom. However when you let a user `Like` an object the options a user has are reduced to two; he either likes it or he doesn't. Although table 3.2 says that when using `Group`, `Score` or `Order` a user can select from `n` categories, and `n` could stand for millions of options, in practice this is usually a small number in the dozen range and therefore we gave those items a Low score. `Insert/Delete` and `Modify` got a medium score because although you are able to come up with your own terms, when `Deleting` items you need to choose from existing items which is choosing from a predefined list and has a low level of freedom. `Inserting` sits on two ends of the spectrum, depending how it is performed: it could have Low level of freedom when you can only insert from a predefined list but has a High level of freedom when you can insert anything you want. When `Modifying` attributes we assume you are somewhat limited by the existing values and cannot replace everything you want.

This analysis will help us in Chapter 5 when we will investigate how we can aid users with making qualitative annotations.

3.2.2 Different tag purposes

After our analysis in the previous section we decided to further investigate tagging, as it was the only annotation type to receive a high level of freedom. These tags can have different purposes of which Gupta et al. have made a categorisation in [5], a summary of this categorisation is shown in table 3.4. From this categorisation we can gather that tags are used for various reasons; annotators not only use it to describe the content but

[2] http://public.oed.com/history-of-the-oed/dictionary-facts/

also to identify the owner or express an opinion.

Type	Description
Content-Based	Identify the actual content of the resource
Context-Based	Identify the context of an object in which it was created or saved
Attribute	Inherent to an object, but can't be derived directly, like author or qualities and characteristics
Ownership	Identifies who owns the resource
Subjective	Express an users opinion or emotion
Organisational	Tags that serve a personal use like a reminder of certain tasks
Purpose	Non-content specific functions that relate to an information seeking need
Factual	Identify facts about an object like people, places or concepts
Personal	Audience of the tag is the author themselves
Self-referential	Tags that refer to themselves
Tag Bundles	Tagging with a link to another tag collection

Table 3.4: Different operation types of tags according to [5].

`Content-`, `Context`-based and objective `Attribute` tags can be considered as a subset of `Factual` tags even though Gupta et al. list them separately. In all three cases the goal is to describe the object as accurately as possible so that it can be retrieved and searched by others. These `Factual` tags are therefore often called descriptive tags. The other main purpose we can identify is `Personal` tags; similar to like we did with `Factual` tags, `Ownership` and `Organisational` can be considered to be a subset of `Personal` tags. These tags won't describe an object but provide additional information which is only of interest for a person or organisation. They can be used to identify who owns the object or even as a TODO note. Because our goal is in the end to let the crowd describe an object the most interesting types of tags are the factual ones. Now that we know how annotations are currently being used in a global manner we will take a look how they are being added to objects in the next section.

3.3 Current crowd annotating techniques

In order to investigate how we can improve the annotation process later in this thesis, we need to look at similar initiatives where the crowd is already allowed to make annotations. By looking at and comparing these initiatives we aim to create a baseline and an overview of techniques when it comes to letting the crowd make annotations. In order to do so we created a list of representative websites which have been featured in literature as state of art with regards to allowing the crowd to make annotations. Most of them are image based, where an image might act as a digital representation of a (CH) item like a vase or brooch. Of the selected initiatives Delicious and Last.fm are the only ones that don't have a focus on annotating images; these have been added to investigate if there is anything different being done in non-image oriented tagging services. Flickr [5][17], Last.fm [5], Delicious [5][17], Picasa [18] have been added because they are prominently used in tagging research. Steve.museum [19], Your Paintings Tagger (YPT) [20] and Accurator [21] have been selected because they have a focus on annotating Cultural Heritage objects.

3.3.1 Comparing Annotation Systems

Now that we have a list of seven user tagging initiatives we need a framework that allows us to compare them so we can investigate theirs strengths and weaknesses. Marlow et al.[22] have provided us with a "taxonomy of tagging systems", which allows us to conveniently compare and analyse each system. They grade each annotation platform on eight different categories which are briefly summarised below:

Tagging Rights Restrictions imposed on group tagging. Options are: *Self-tagging*; users are only allowed to tag resources they created. *Free-for-all*; users can tag any resource. *Permission-based*; certain users can be allowed to tag certain objects based on permission levels.

Tagging Support The mechanism for tag entry. Options are: *Blind tagging*; a user cannot view tags assigned to the same resource by other users. *Viewable tagging*; the user can see the tags already associated with the resource. *Suggestive tagging*; the system suggests possible tags to the user.

Aggregation Model The aggregation of tags around a given resource. Options are: *Bag-model*; a multiplicity of tags for the same resource which may result in duplicate tags from different users. The bag model allows for aggregate statistics. *Set-model*; users collectively tag an individual resource, denying any replication.

Object type The type of the resource being tagged. This consists of a broad range of options. Examples are: *web pages, bibliographic material, blog posts, images, users, video or audio*. These can be divided in *textual* and *non-textual resources*.

Source of Material How the resource to be tagged was added to the system. Options are: *User-contributed*; users can upload their own material. *System*; the resource is given to the users by the system. The resource is usually added by a system administrator. *Global*; the system is open for tagging any web resource.

Resource Connectivity Resources can be linked to each other independent of the user tags. This can be useful when for convergence of similar tags in *suggested* and *viewable* scenarios. Options are: *Linked*; ex. web pages are connected by directed links. *Groups*; resources can be placed in groups. *None*; resources aren't linked to each other.

Social Connectivity How users within the system are linked to each other. The options are similar to *Resource Connectivity*.

Based on this classification system we made an overview in table 3.5 which compares the previously mentioned tagging systems.
From table 3.5 we can see that systems where the users provide the taggable resources themselves they are usually limited to self-tagging, whereas if the resources are provided by the system users are allowed to tag them all. This is because the goals of the annotations are different. When images are provided by the system, they are

Site	Tagging Rights	Tagging Support	Aggregation model	Object type	Source of Material	Resource Connectivity	Social Connectivity
Flickr	Self-tagging	Viewable & Suggestive	Set	Images	User	Groups	Linked
Picasa	Self-tagging	Blind	Set	Images	User	Groups	Linked
Steve.museum	Free-for-all	Viewable & Suggestive	Bag	Images	System	None	None
Your Paintings	Free-for-all	Blind & Suggestive	Bag	Images	System	None	None
Accurator	Free-for-all	Blind & Suggestive	Bag	Images	System	None	None
Last.fm	Free-for-all	Viewable & Suggestive	Bag	Audio	System	Groups	Linked
Delicious	Self-tagging	Blind	Bag	Web pages	Global	Groups	Linked

Table 3.5: Categorisation of annotation platforms according to [22].

provided so that other people can tag it in order to help the provider of the images. When a user uploads his own images he usually doesn't need any help as the volume of uploaded objects is smaller and because, in the case of images, it usually is his own creation and therefore he has the best knowledge about the objects. Although Marlow et al.[22] state that for *Tagging Support* there are 3 distinct categories, namely *Viewable, Suggestive* and *Blind*, we found there are some exceptions. Although Viewable and Blind are mutually exclusive, we often came across a combination of Viewable or Blind together with Suggestive tagging. It is therefore that this column often has two values in table 3.5. Of the image related sites we see that with the CH initiatives Blind & Suggestive tagging dominates. This is probably because these institutes want to have a diverse range of tags and therefore try to prevent users being influenced by others as they want the users own interpretation. It is likely that therefore the connectivity aspect plays a small to non-existent roles on these sites. Most systems use a way of suggesting tags, as this helps with standardising the tags used in the folksonomy. Furthermore, most systems use the bag aggregation method, as this is useful for metrics. By using this model we can see which tags are used more often and therefore determine which ones are deemed more important than others.

When looking at the CH initiatives we can see that they are almost the same. The only difference is that Steve.museum has Viewable tagging support instead of Blind tagging as used by YPT or Accurator. Because the three initiatives are basically the same, we conclude that these features are the minimum a CH annotation system should provide.

3.3.2 The Annotation Process

Now that we have an overview of different annotation systems, it's time to look at the process. The process of making an annotation always starts with the studying of the object. In the case of an image this means that one must carefully look at what is depicted but in other cases this might mean listening to an audio recording or reading a website. After the object has been studied, the user can add a tag. All the websites have at least one input field where tags can be added. Multiple tags can often be added at once by separating tags with commas or spaces. Accurator and YPT both provide multiple annotation fields which represent different tasks a user should perform. For example YPT has 4 input fields for *Things*, *People*, *Places* and *Events* [20]. The reason for this is twofold: 1) They help direct annotators with focusing on specific elements of interest in an object and 2) they allow for different suggestion methods per input field.
When items are being dynamically suggested, the author can select a suggestion, or choose to redefine his input in order to find another suggested term. After the user is satisfied with his input, he can then press enter and submit the term(s) to the system, adding it to the resource. This process has been detailed in Fig. 3.1.

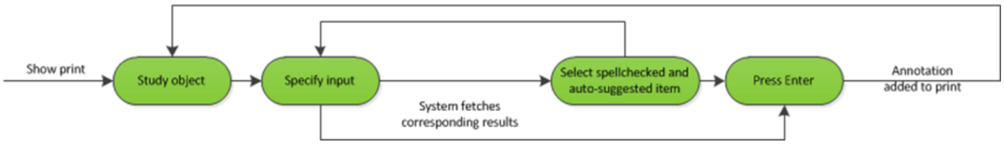

Show print
Study object
Specify input
Select spellchecked and auto-suggested item
Press Enter
Annotation added to print
System fetches corresponding results

Figure 3.1: A generic annotation process.

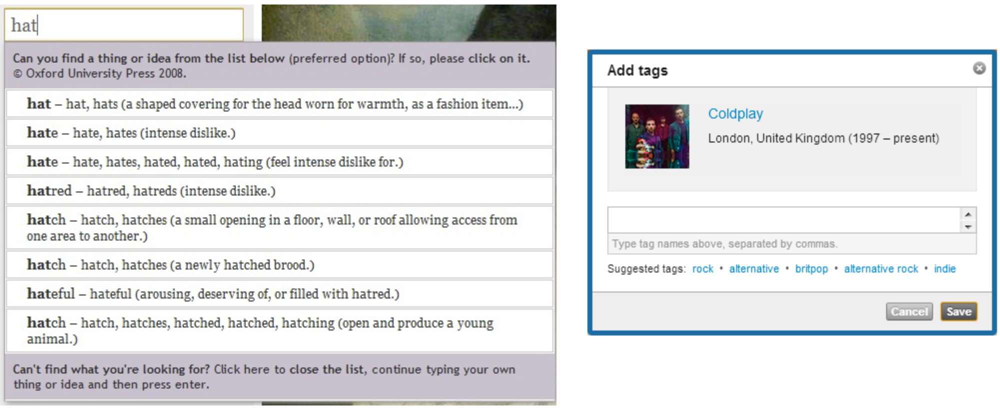
Site	Study object	Input	Suggestion	Submission
Flickr	Yes	Free text	Yes*	Click / Enter
Picasa	Yes	Free text	No	Enter
Steve.musem	Yes	Free text	Yes*	Enter
Your Paintings	Yes	Free text	Yes	Click / Enter
Accurator	Yes	Free text	Yes	Click / Enter
Last.fm	Yes	Free text	Yes*	Click / Enter
Delicious	Yes	Free text	No	Enter

*Suggestions come in the form of static predefined lists.

Table 3.6: An overview of annotation options on representative websites

In order to verify the process we have analysed for each website mentioned in the previous section if the process is able to represent it. This verification can be seen in table 3.6. Each website contains an object which needs to be studied before making an annotation. The only exception is Delicious where the annotatable object isn't contained in the website because with Delicious you annotate (a link to) another website. However, this website must first be studied before one can make any annotations. All websites allow *Free text* as input. This means that the user is able to type whatever he likes and isn't restricted in any way. When a site makes suggestions, they are often made in different ways where we can make a distinction between *static* or *dynamic* *suggestions*. Dynamic suggestions are being made based on the input by the user, for example by using auto-complete or auto-suggest, as used by Google's search engine. In the case of YPT terms are being suggested from the Oxford English Dictionary [20][3]. Static suggestions are shown before a user has done anything at all and can be based on previous input by others or themselves.

Based on the input of free text, the ability to disregard suggestions and submit any free text string that is desired, the assumption has been made that all systems store the made annotations in a folksonomy.



hat

Can you find a thing or idea from the list below (preferred option)? If so, please click on it.
© Oxford University Press 2008.

- **hat** – hat, hats (a shaped covering for the head worn for warmth, as a fashion item...)
- **hate** – hate, hates (intense dislike.)
- **hate** – hate, hates, hated, hated, hating (feel intense dislike for.)
- **hatred** – hatred, hatreds (intense dislike.)
- **hatch** – hatch, hatches (a small opening in a floor, wall, or roof allowing access from one area to another.)
- **hatch** – hatch, hatches (a newly hatched brood.)
- **hateful** – hateful (arousing, deserving of, or filled with hatred.)
- **hatch** – hatch, hatches, hatched, hatched, hatching (open and produce a young animal.)

Can't find what you're looking for? Click here to close the list, continue typing your own thing or idea and then press enter.

Add tags

Coldplay
London, United Kingdom (1997 – present)

Type tag names above, separated by commas.

Suggested tags: rock • alternative • britpop • alternative rock • indie

Figure 3.2: Dynamic vs. static suggestions

3.4 Quality

Now we know how annotations are created, let's take a look at the resulting quality of these annotations. In Xu et al. [17] they list the following criteria for tag quality:

High coverage of multiple facets A good tag combination should include multiple facets of the tagged resource.

High popularity If a set of tags is used by a large number of people it is less likely to be spam. This means that the tags should have a high frequency.

Least-effort The number of tags for identifying an object should be minimised and the number of objects identified by the tag combination should be small.

Uniformity (normalisation) Due to a lack of a taxonomy or ontology, Xu et al. notice two types of divergence in tags: divergence due to syntactic variance: *blogs*, *blogging* or *blog* and those due to synonyms *cell-phone* and *mobile-phone*. The divergence has pros and cons. It introduces noise to a tagging system, but it also can improve recall. Their proposed solution is to let the users free but collapse the variances to an internal canonical representation.

Exclusion of certain types of tags Tags which are entered for personal use are less likely to be used by other users and therefore should be excluded.

[3] http://www.oed.com/

When we look at the quality of the tags of the systems seen in the previous sections, we can only look at the sites where we can see the tags, meaning their *Tagging Support* method should have *Viewable tagging*, which are Flickr, Steve.museum and Last.fm. During our analysis of existing systems we however attended presentations about the YPT and Accurator projects and therefore we can hopefully make some informed assumptions about their quality. Here we have chosen once again to use a 3-point Likert Scale as this should provide sufficient granularity.

Site	Coverage	Popularity	Least-Effort	Uniformity	Exclusion
Flickr	High	N/A	Low	Low	Low
Last.fm	High	High	Low	Medium	Low
Steve.museum	High	High	Low	Low	Low
Your Paintings	High	High	Medium	Medium	Medium
Accurator	High	Low	Medium	Medium	Medium

Table 3.7: Tagging sites rated according to [17]

When we look at Flickr we see that due to the low restrictions a large part of the object has been annotated. But unfortunately the terms are often mixed in languages and in spelling, as can be seen in figure 3.3. Users also use multiple tags to identify the same concept, making it score low on uniformity. It is also not uncommon for photo's to be tagged by users with the type of camera the picture was taken with, which is redundant because this information is pulled from the EXIF[4] metadata and users can therefore browse pictures based on camera category[5]. For the reasons that this is data that doesn't belong there and the fact that users have a tendency to over tag, Uniformity and Exclusion also receive a low score. Unfortunately Flickr tags can only be added by a single user, so we can't say anything about the popularity of the tags for that resource.

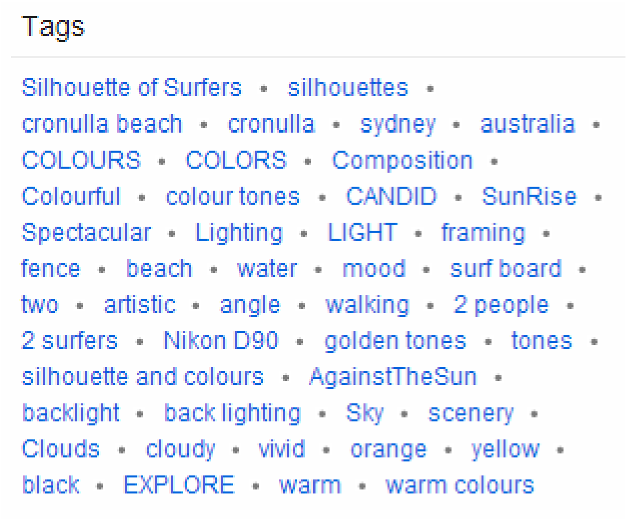
Last.fm tags also have a high coverage, listing various aspects of a song or artist. When looking at the tags of a resource we noticed that each object has exactly 60 tags which are the likely the top 60 added tags for that resource. A subset of those tags is also listed in a larger typeface to display which are the most popular. Even though we only see the top sixty of tags, it becomes clear that even then we see a lot of tags referencing to the same concept. As can be seen in figure 3.4 using a popularity vote on tags does not guarantee that the quality of the tags will be high as even then things can be manipulated. Because Last.fm suggests tags for users, they score a medium on uniformity. Steve.museum works mainly the same as Last.fm, and therefore the scores are quite similar.

From a presentation by Your Paintings Tagger [6] we know that they pick the top x of most added tags by various users and only use these tags when feeding the user generated content back into an institutes system. Because they ask directed questions *(What things do you see?)* the most important topics of interest are covered and suggestions based on a dictionary help with obtaining a focused set of annotations. When we compared the different annotation systems in section 3.3.1 we already noticed that

[4] http://en.wikipedia.org/wiki/Exchangeable_image_file_format

[5] http://www.flickr.com/cameras/sony/slt-a55v/

[6] Presentation at Erfgoed Delft, May 2012



Tags

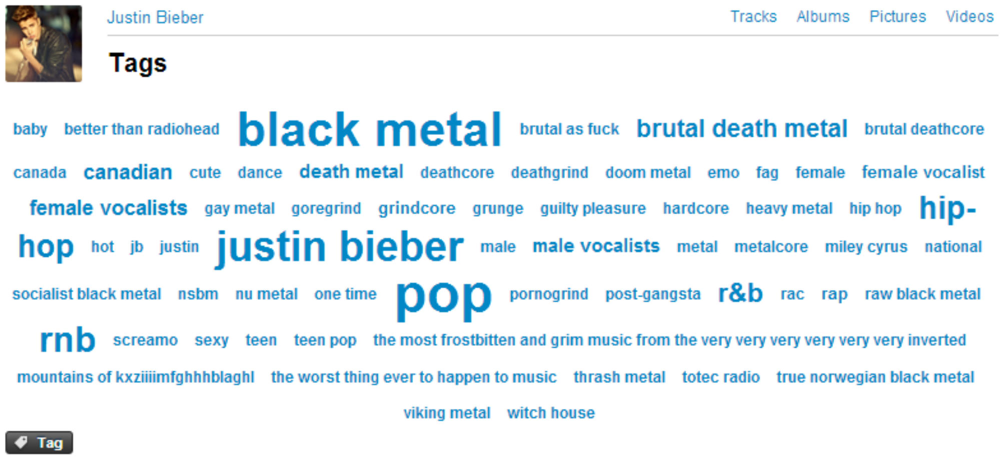
Silhouette of Surfers • silhouettes • cronulla beach • cronulla • sydney • australia • COLOURS • COLORS • Composition • Colourful • colour tones • CANDID • SunRise • Spectacular • Lighting • LIGHT • framing • fence • beach • water • mood • surf board • two • artistic • angle • walking • 2 people • 2 surfers • Nikon D90 • golden tones • tones • silhouette and colours • AgainstTheSun • backlight • back lighting • Sky • scenery • Clouds • cloudy • vivid • orange • yellow • black • EXPLORE • warm • warm colours

Figure 3.3: Over tagging on Flickr.

Accurator scored similar to YPT feature wise and quality wise it's much the same. The only difference is that their tags score low on popularity; the main reason for this is that it currently still has a small user base. When using a small set of users it's less likely to find a match in similar terms and therefore their tags have a low popularity.

3.5 Conclusion

In this chapter we looked at the different operation types of how annotations could be added to an object, the different purposes of these annotations and how they are being used in a CH institute. When comparing different crowd annotation platforms we saw that most use a folksonomy to structure their tags and spotted that most CH initiatives



Tags

baby better than radiohead black metal brutal as fuck brutal death metal brutal deathcore canada canadian cute dance death metal deathcore deathgrind doom metal emo fag female female vocalist female vocalists gay metal goregrind grindcore grunge guilty pleasure hardcore heavy metal hip hop hip-hop hot jb justin justin bieber male male vocalists metal metalcore miley cyrus national socialist black metal nsbm nu metal one time pop pornogrind post-gangsta r&b rac rap raw black metal rnb screamo sexy teen teen pop the most frostbitten and grim music from the very very very very very very inverted mountains of kxziiiimfghhhblaghl the worst thing ever to happen to music thrash metal totec radio true norwegian black metal viking metal witch house

Figure 3.4: Manipulation of tags on Last.fm.

have a similar feature set. Whilst the coverage of most objects is sufficient there is room for improvement with regards to the uniformity and the number of tags needed to describe a resource. In the next chapter we will see how a CH institute, in our case, the Rijksmuseum in Amsterdam tackles these problems and in chapter 5 we try to use the lessons learned in this chapter and set the outline for our system.

Chapter 4

Use Case
Rijksmuseum Amsterdam

In this chapter we will look at how professionals in a Cultural Heritage setting make annotations. For this we will look at *The Printroom Online* project at the Rijksmuseum in Amsterdam (RMA).

4.1 About The Printroom Online

The print collection[1] of the RMA is considered to be the largest and best print collection in the world. The collection consists of nearly 700.000[23][2] prints, drawings and photographs and encompasses prints from a period of the 15th century till today. Until 2007 the only way to view these prints was either to wait until they've became available in an exposition or to view them in the RMA study room. Therefore the Print Room Online initiative was started; it's main goal is to realise the registration, annotation and digitisation of the print collection[21]. The project consists of registering basic properties of each print such as location, object number, maker, sub-collection, measurements and location. In order to make the collection available on-line to the public each object will be photographed and get subject matter annotations (see page 12 for the definition of Subject Matter).

4.2 Current Process

The Printroom currently consists of six cataloguers, a photographer, a curator and a project manager [3]. The cataloguers are highly educated professionals, each with knowledge of a particular part of the domain, and provide the objects with their basic annotations. In order to start the process a cataloguer selects a print from a box. Each print is stored in a box, ordered by creator, and these boxes are processed in alphabetical order. Every cataloguer tries to process 25 prints each day and therefore spends on average 20 minutes in processing per print. The process as described in [21] is as follows:

[1] https://www.rijksmuseum.nl/en/explore-the-collection/works-of-art/prints

[2] Source: Meeting with various employees of the Printroom Online Project

[3] The processes and situations described here are as they were in early 2012

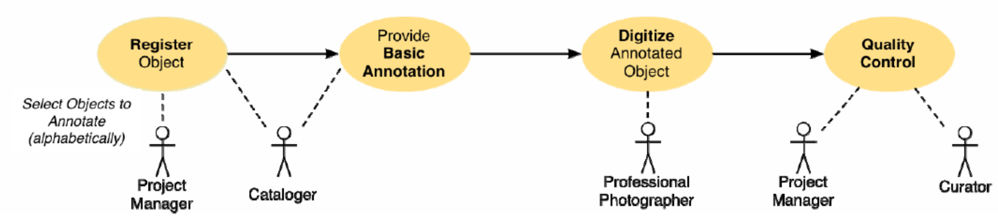


Figure 4.1: Print Room Online project workflow. From [21].

1. New object has been acquired and is processed for storage
2. Added to the proper collection
3. Signal sent to annotators notifying them of the new object and location
4. Annotating
5. Digital photograph
6. Manual check by collection manager

After an object has been processed for storage the cataloguers register each object of the box in the collection management system by entering its information from the RMA inventory books. This is mandatory information like the creator, the name of the object, acquisition date and method. Because each box is sorted by creator, the annotators process multiple prints by the same creator in a row. When starting with a new creator, each creator needs to be thoroughly researched. The annotators do so using various art-historic databases like RKDartists&[4] or by using relevant books. Each source of information is added to the system as reference.

When an object has been registered the next step is adding annotations. As can be seen in table 3.1 in the previous chapter an annotator needs to fill in many categories and describing the print is only one of the 31 fields that need to be added. Because table 3.1 is a standard proposed by Getty it doesn't mean that the RMA completely uses the same list of fields, but the gist is the same; subject matter annotations are only a small part of the process. Fortunately the other categories are quite standardised in their ways of being processed and the guidelines for the basic registration are quite straight forward. For example, it's not that complicated to annotate the size of an object, but you do need to agree upon a standard the unit for storing the measurements. Unfortunately the situation isn't the same for subject matter annotations. In the CH community there is little consensus and limited tooling support for subject matter annotations[23]. The RMA therefore decided to focus on the *person* (who), *event* (what), *location* (where), *date* (when) and iconography (See fig 4.2). To save time and achieve the desired throughput rate of 25 prints a day the annotators limit themselves to the main theme depicted on the object. After a box has been completely processed by a curator, the box is sent to a professional photographer and the entire box gets digitised. The photographer processes roughly 150 objects a day and each object is

[4] http://www.rkd.nl/rkddb/

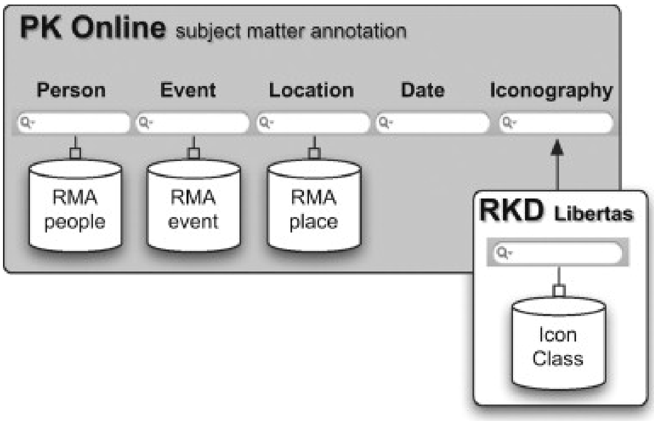


Figure 4.2: Sketch for the RMA setup for subject matter annotation. The person, event and location fields give access to an associated, internal thesaurus. Date is a free text field without a thesaurus and Iconography uses an external thesaurus called Iconclass which codes are manually copy pasted into the Iconography field. From [23].

stored in different resolution sizes. As a final step in the process the project leader will perform the majority of the Quality Control. For a institution like the RMA the quality and correctness is of the utmost importance. The project leader will check if all the fields are filled in according to the in house rules and regulations and also take the correctness and consistency into account. The project leader spends around 8 hours a week on Quality Control and processes nearly 150 objects.

4.2.1 Iconclass

The iconography field at the RMA uses the external taxonomy called Iconclass[5]. Iconclass is a classification system designed for art and iconography. It is a taxonomy which consists of definitions of objects, people, events and abstract ideas and is maintained by the RKD[6]. The system stores over 28.000 definitions, divided in ten main categories. Each definition consists of an alphanumeric notation and a textual description.
Although there isn't a specific guideline within the RMA for annotating with Iconclass codes, they try to annotate the following main themes:

1. Bible
2. Personification
3. Mythology
4. Symbolic

When trying to create subject matter annotations for a print when selecting Iconclass codes the curators have no further rules defined. They mainly try to use common sense

[5] http://www.iconclass.org/

[6] http://www.rkd.nl/

and add no more than 5 key elements. They won't annotate common entities like glasses or shoes; they only do so when they stand out. This means that those entities only will be added when they are either remarkable for that period or are mentioned in the description of a print. Adding an Iconclass notation is done by browsing the Iconclass website and manually copy and pasting the annotation codes in the designated field in the RMA collection management system. Objects can be found via their Iconclass codes, so they can see which other objects have the same code, which is useful to check for correctness.

When Hildebrand et al. [23] studied the RMA's annotation process they noted that most of the RMA's annotations are being made using the Iconclass taxonomy. Later on they note when they try to use heterogeneous thesauri that Iconclass can also be used to answer the *Event/What* question in the annotation process.

	People	Place	Event	Iconclass	Total
#	9245	9034	6509	30 981	55 796
%	17	16	12	55	100

Table 4.1: For each thesaurus, the total number of terms used for the annotation or the RMA objects and the percentage relative to the total number of terms is used. Created in 2009 after 30 747 prints had been annotated. From[23]

You *could* also use Iconclass to describe the persons and locations, be it in a limited fashion. This is supported in a couple of ways:

Free Text

Iconclass allows a small level of freedom when using their annotation codes. In Iconclass you find can notations like *61E(...).* Because Iconclass codes are manually copied and pasted into the collection management system, the dots can be freely filled in by the curator, for example *61E(Bavel).* Due to the fact that we have 61E before the parentheses we still know that the text placed in between represents a, in our case, name of a city or village. Unfortunately this practice is contradictory to a true taxonomy and therefore should be avoided as much as possible.

Event based

Iconclass has elaborate descriptions. For example:

71I313 - Solomon returns to Jerusalem

Although this is an event, it already describes the who and the where. Therefore when annotating with Iconclass it is often not needed to list the person and location separately when one uses a precise event description.

Because a taxonomy usually has a specific focus; other thesauri could be used to annotate persons and locations. Hildebrand et al. did so by using DBpedia persons in combination with Getty ULAN for persons and Getty TGN for locations.

4.3 Area's for improvement

When studying the RMA process in the previous section we have identified room for improvement in three key areas: *Time*, *Quantity* and *Expertise*. Each will be detailed in the sections below.

4.3.1 Time

An important keyword from the previous sections is time. Each cataloguer tries to spend no more than 20 minutes on a print as they need to keep moving. Therefore they limit themselves making no more than 5 annotations per print as otherwise it would be too time consuming. Only a small portion of the prints then get verified as the project manager doesn't have the time to verify all the processed prints. As mentioned in the previous section, a project manager only has 8 hours a week for checking Iconclass codes. This results in about 150 checked objects per week. The problem here is however that the 6 curators together annotate 150 objects *per day*. Furthermore, it will still take a couple of years before the project has been completed. When we try to extrapolate the 150 objects a day to a year we get an average number of 38.250 prints annotated per year. This means that with a set of 348.000 annotation-less prints, which is a number that increases every day as the collection grows, it will take nearly a decade before the complete collection has been annotated. Because these are highly trained professionals and professionals are usually expensive this problem isn't simply solved by hiring more professionals.

4.3.2 Quantity

Because the cataloguers only have time to add no more than five annotations to each print, the number of subject matter annotations per print is relatively low. When we query our copy of the RMA collection database[7] we get an average of 0,29 annotations

Number of Annotations	Number of Prints
0	348.004
1	35.162
2	14.306
3	6.897
4	3.373
5	1.360
6	654
7	400
8	310
9	240
10	166
>10	196

Table 4.2: Annotations per Print at the RMA as compiled from the dataset

[7] Copy obtained in early 2012 as part of the COMMIT program

per print (Out of 411 thousand prints only 63 thousand prints have annotations). When we then refine our search and only look at prints which actually have annotations, we get an average of 1.91 annotations per print. Out of 63.064 prints with annotations, only 1966 have more than 5 annotations. For a complete overview we refer you to table 4.2. When we want to increase the number of average annotations per print, the cataloguer needs to spend more time processing each object and therefore the project will take even longer.

The quantity is also reflected when we look at the quality scale by Xu et al. [17]. We can see that the RMA performs quite well when compared to other sites mentioned in table 3.7 but due to the fact that the average tag per print is 1.9 the coverage is rated as low. The tag that is added is however usually short and precise, containing already a lot of information, making it score high on Least-effort. Because the annotations are made with the Iconclass taxonomy the uniformity and exclusion of certain tags score high, as one can only select from a limited number of tags. Popularity isn't rated, as each print is only annotated by a single person.

Coverage	Popularity	Least-Effort	Uniformity	Exclusion
Low	N/A	High	High	High

Table 4.3: Quality of the RMA's annotations when rated on the scale by [17]

4.3.3 Expertise

The ten cataloguers each have their own areas of expertise. But unfortunately it is impossible for them to have enough expertise in all the domains covered by the prints in the collection [21]. Examples are castles or botanical elements. Although the curators can perfectly well say that something is a castle or a tree, it is harder for them to define which castle it is or what type of tree it is. In order to overcome this problem, when available, the RMA will consult external experts to help them with certain domains they lack knowledge.

4.3.4 Conclusion with regards to problems

In conclusion we can say, that most problems with regards to *Quantity* and *Quality* are caused by the lack of time. The curators can't do everything they want simply because there isn't enough time or manpower to do the task. Because there is a small group of annotators some domains remain unexplored because the knowledge isn't available within the organisation.

Chapter 5

Supporting Crowd Annotators

In the previous chapters we described how the crowd and professionals currently annotate and noticed that there is room for improvement where both parties can learn from each other. In this chapter we will discuss how we can improve the current situation and describe processes to do so in a qualitative way.

5.1 Crowd vs. Professionals

When we combine tables 3.7 and 4.3 from our analysis according to Xu et al. [17] of the crowd initiatives and the professionals in table 5.1, we see that the professionals are strong where the crowd is weak and vice versa. Crowd initiatives have a high coverage but the individual tags are of lower quality, whereas at the RMA the tags are of high quality but the coverage is low due to time constraints. What we would like to get is the best of both worlds. When one uses the Iconclass taxonomy you get the Least-effort, Uniformity, and Exclusion characteristics for free with the trade-off that personal tags aren't possible anymore. Because a print at the RMA is only tagged by one person, the project manager needs to check the quality of the tags (although experts hardly need correction). By letting multiple users annotate the same object we hopefully could increase the Coverage and Popularity characteristics, obtaining a broader disclosure of the print with a higher likelihood that the most popular annotation is the correct one.

Site	Coverage	Popularity	Least-Effort	Uniformity	Exclusion
Flickr	High	N/A	Low	Low	Low
Last.fm	High	High	Low	Medium	Low
Steve.museum	High	High	Low	Low	Low
Your Paintings	High	High	Medium	Medium	Medium
Accurator	High	Low	Medium	Medium	Medium
RMA	Low	N/A	High	High	High

Table 5.1: Combination of tables 3.7 and 4.3

It becomes clear that the crowd could be a valuable addition to the process of the RMA because both could likely enhance each other. The major strength of the RMA is that its catalogues use a taxonomy for annotation. Therefore what we would like to investigate is whether or not the crowd is able to annotate CH objects with a taxonomy. Another aspect is that the demographics of a group of crowd annotators can vary widely. This group can be comprised of fisherman, housewives or collage students but there is also a possibility that an expert from another CH institute is among them. Although the expert will likely do just fine we might need to look at methods how we can support the crowd and bring their knowledge to a sufficient level to annotate these objects.

5.1.1 Crowds and taxonomies

When looking at the crowdsourcing initiatives in Chapter 3 all of them used a folksonomy. Unfortunately we were unable to find any crowd initiatives that use a taxonomy, so although there is slight chance they exist, they are certainly uncommon. Furthermore we spotted a trend in tagging literature where most researchers praise the open structure of folksonomies and deem taxonomies too rigid for regular users[5][24]. However in table 5.1 we've clearly seen that there are advantages to a taxonomy with regards to quality and even a CH institute like the RMA favours the Iconclass taxonomy over a folksonomy. Therefore we have identified a couple of potential obstacles which could occur if the crowd should use a taxonomy, which we detail in the sections below.

A taxonomy has a limited vocabulary

Due to the limited vocabulary lay annotators can express themselves less freely, as they can only pick from a finite set of terms. Because a taxonomy is maintained by a group of experts of (usually) an external organization it is harder to introduce new terms. This is one of the reasons that taxonomies have difficulties with keeping up with the increasing and evolving vocabulary people come up with every day. We performed a small experiment where we let people search for terms using the Iconclass taxonomy and most could find the majority of the terms they were looking for, although some were described a different manner. For example; it doesn't contain the tag *"lady"* but does contain *"female sex; woman"*. Because a taxonomy is mainly used by a professional some terms might be different then what a crowd annotator is used to. For example, in some areas of study it is common to use the Latin name to represent an entity and is as such represented in a taxonomy. As an example in the CH sector the director of the YPT project noticed in [20] the following differences in phrasing between the experts and the crowd annotators: *The term 'abstract' tends to get applied by taggers to any slightly modernist treatment of a subject, rather than purely, (. . .), non-representational paintings; and the term 'portrait' is applied to depictions of biblical, mythological or symbolic human figures such as Jesus Christ,(. . .), in addition to the conventional and more specific definition.*

Using a folksonomy can be favourable as it doesn't limit the user's ability to express themselves and the way people annotate is likely similar to the way they search. The downside however is that people can annotate objects using tags like *"silly hat"* or *"really large castle"*. Another negative of free expression is that people can make accidental spelling mistakes and as a result objects will end up with tags like *"catsle"* which can't be used for retrieval and does not refer to a concept. The YPT and Accurator systems try to prevent this by suggesting words and titles from word lists like Wikipedia article titles and/or various dictionaries [20].

A taxonomy is skewed towards a certain topic

A vocabulary is usually skewed towards a particular topic. This is caused by the limited number of editors that create and maintain a taxonomy and therefore might also represent their views and cultural biases. Iconclass has a large focus on Christianity and contains a lot of events from the Bible but none from the Koran. It would therefore be difficult to annotate objects from this religion with the taxonomy. The Iconclass taxonomy consists of ten categories; five main categories to describe all the principal aspects of what can be represented and four categories to accommodate 'special' topics which are more narrative of nature with an emphasis on the Bible and classical mythology. The tenth and last category exists to represent abstract art [1].

Although the specialisation in a specific topic is deemed difficult for users, it is also a strength of a taxonomy. It takes one topic and provides a complete overview of it instead of a small overview of many topics. And because a large part of the RMA consists of prints with a biblical background the Iconclass taxonomy is a proper fit.

A taxonomy has a hierarchical structure

A taxonomy has a clear hierarchical structure with predefined categories. Due to this rigid structure it might be hard to place a term in a specific category as it can possibly fit multiple. The result of this is that when browsing the taxonomy it can be hard to make the correct choices of where a term might reside. However the structure is very useful in the case when a found term is too specific or not specific enough. In that case we can use the treelike structure traverse a level for a more specific or broader term.

5.1.2 Converting folksonomies to taxonomies

From the previous three sections it has become clear that there are still some issues before a non-expert user can use taxonomy. There are studies whose aim is to map a folksonomy to a taxonomy [2], although most literature focuses on creating a new taxonomy from folksonomy data [13][14][25]. We have however decided not to pursue this route for the following reasons: 1) By doing so we have created a new problem. New errors get introduced into the system as mapping free text to a taxonomy term isn't a trivial task. You have to deal with ambiguous words which can have a different meaning based on the context and synonyms which need to be mapped to a single term. This conversion step can then have an impact on the quality of the annotations. 2) We want to stay as close to the task of the professional as possible as their method has proven itself as able to provide qualitative annotations suited for their needs.

Therefore we think the real challenge lies in letting non-professionals use a taxonomy. As a result the annotations will be in the same format as the professionals and therefore it should be more convenient to let these flow back into the professional process as discussed in section 4.2.

[1] http://www.iconclass.nl/contents-of-iconclass

[2] http://hugh.thejourneyler.org/2012/from-folksonomies-to-taxonomies-with-linguistic-metadata/

5.1.3 Crowds vs. Knowledge

A professional isn't called a professional because their ability to use a taxonomy, their main expertise lies in their background and education which is likely in the form of an art degree. The chance that a crowd annotator has an art degree is relatively low. Although an art degree isn't necessarily needed to describe what's portrayed on the object, as the annotator can see what's depicted on the object, a proper background could aid them making better annotations. The more a user knows about an object, like the a (historical) context or figurative aspects of a work, the better an annotation should get. An example of this could be figurative aspects of an object; when looking at Biblical prints God can be represented in many different shapes and sizes, from an all-seeing eye to a hand (Dextera Dei). Therefore users should be educated in recognising these aspects.

5.2 Supporting crowd annotators

In the previous section we outlined difficulties a non-professional user might have when using a taxonomy to annotate objects. In order to help a user overcome these difficulties we've identified the following areas for improvement; we would like to help a user with ...

1. ... finding the most suitable term for a print
2. ... navigating through the structure
3. ... learning about a object

In order to aid a user with these three steps, we will revise the generic annotation process outlined in chapter 3 in section 5.2.1 and create different strategies to help solve these three issues in section 5.2.2.

5.2.1 The Taxonomy Annotation Process

In Chapter 3 we have identified the generic annotation process. For this process we have identified four steps for when a print is shown to a user: 1) A user studies object and determines what he wants to annotate. 2) The user specifies his input or keywords describing the item to be annotated in an input field. 3) When the system returns a set of results the user evaluates the options which are presented to him. 4) If a suitable option has been found the user clicks on the annotation which attaches the annotation to the print or the user ignores the options and submits his own term. It is important to note that in some scenarios step two might not need to be executed by the user, but by the system. In this case instead of keywords the system will take a print or an already known term attached to an object as input and will return a set of results based on that information. Our goal is to modify this process so that it is suitable for a taxonomy.

When using a folksonomy like approach users are allowed to submit their own terms whereas they are unable to do so in a taxonomy setting. The user must select a term from a predefined set present in the taxonomy. As a result, the option to bypass the suggestions isn't available anymore, whereas the rest of the process can remain similar to what we defined in the generic annotation process. Although we've only eliminated

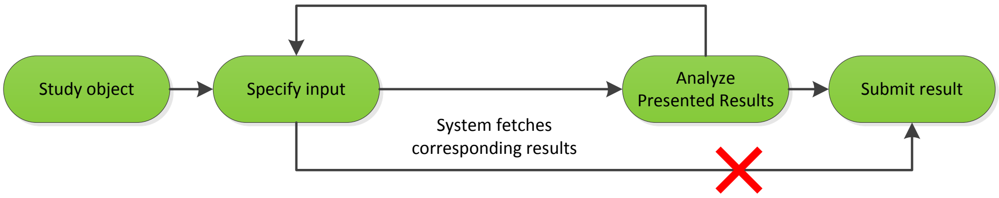

Study object
Specify input
Analyze Presented Results
Submit result
System fetches corresponding results

Figure 5.1: The annotation process as seen in fig. 3.1 for use with a taxonomy. The user is no longer able to bypass the presented results as he is forced to pick one.

a single arrow, as seen in figure 5.1, the perspective has changed drastically. We have moved from a perspective where a user adds a tag and suggestions are being made, to a perspective where a user searches in the taxonomy for the tag that relates to his input. This is also represented in the first of our three areas of improvement, as stated in the introduction of this section: "the user needs to *find* the most suitable term".

5.2.2 Strategies for aiding a user

In order to help a user with find terms we studied techniques which are commonly used in search and recommendation systems. Based on the input a user or the system, there are various strategies which can be used to compile a list of terms from the taxonomy. Based on figure 5.1 we can identify four different categories:

1. The Input layer Processes the input received from the user.

2. The Retrieval layer - Generates a list of terms from the taxonomy based on the input.

3. The Presentation layer - Presents the taxonomy items to the user in different ways, for example sorting or providing additional information.

4. Submission layer - Different ways of submitting the selected term to the system.

Although this process only consists of a limited number of steps, this does not mean that the strategy options for each layer are limited as well. For each part in the process there are multiple of ways to implement each step and these steps can be combined in different ways. We analysed search based systems like Google[26], Amazon[27] and based on this analysis we created a collection of options which can be used to create a strategy. This collection is depicted in figure 5.2. It is important to note that this is not a full list and we are not limited to these options. The image shows that even with a relatively small set of options we could quickly get a lot of different implementations of the taxonomy annotation process.

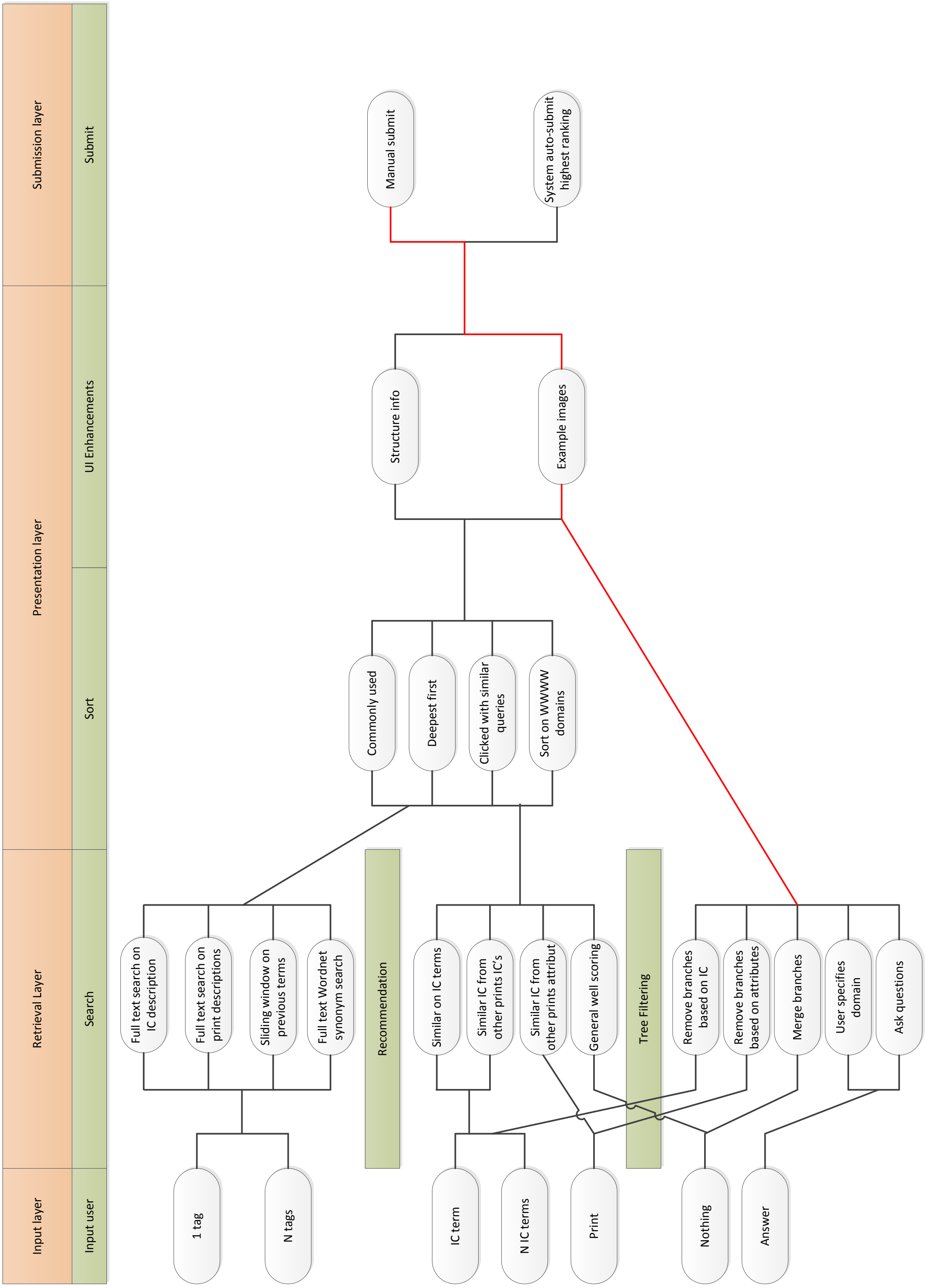


Figure 5.2: Possible implementations of the System Perspective.

In the `Input` layer there are three sections: 1) we use the input provided by the user; the search query or an answer to a question. Or 2) we use information we already know, and take information about a print or other annotations as input for our strategy. And finally 3) we use no input at all, which can be used for generic strategies.

In the `Retrieval` layer a distinction has been made between different categories of techniques for obtaining a list of terms. The *Search* category is based on Information Retrieval techniques, where the system tries to find matching terms based on a search query. In the *Recommendation* category, based on Recommender systems, the system tries to recommend terms based on, for example prints similar to the one which have been received as input. This is similar to what Amazon does with recommending books and CD's based on previously browsed objects. Finally there is the *Tree Filtering* category, which manipulates the taxonomy and breaks down the tree or reorders it (which can cause that the tree will be no longer a tree) into something which aims to be easier to use.

In the `Presentation` step a distinction has been made between a *sorting* and a *UI enhancements* step. The *sorting* step can rearrange the taxonomy terms in a particular order, providing a different ranking for the terms found in the previous step. Even though various search algorithms already have a ranking system, which usually tries to rank the found terms on which resembles the most to the search query, it is often also useful to sort based on attributes of the found data. A common example of this can be seen in webshops where one can sort different products based on price. An example in our case this could be the frequency of which a term has been used by professionals. The *UI enhancements* step adds additional information to the User Interface, usually by giving the user more information that allows him to make a better assessment of which term to choose. This can for example be done by adding images of prints to all the annotations in the list that contain the same annotation. The UI enhancements can also be used to provide the user with extra knowledge as mentioned in section 5.1.3.

In the `Submission` layer we show two options of how an annotation gets added to an object. This can be done manually in various ways by the user, for example by clicking on an item in the list, which is the common use case. Other examples could be that a user selects a region on an object and attaches the annotation to it, so we know exactly where subject that has been annotated is located on the object.

What is important to notice here is that, unlike the Iconclass browser mentioned in the previous chapter, in various strategies we make use of the data already known by the system. These can be annotations already made by professionals or information about a print, like the author. By leveraging the data we already have and more importantly, know is correct and of high quality, we can help the user and provide him with more guidance when looking for the best fitting annotation for the object he is trying to describe. By employing different strategies we aim to make a complex taxonomy more accessible to a lay user.

5.3 Use case

Given the large number of possibilities we have for implementing a strategy to aid a user, it becomes clear that there are simply too many for the scope of this thesis. Therefore we cannot investigate if a specific recipe would yield the best results and should be used. But even if we had the time to do so, it is highly unlikely that such a task exists. Different goals and datasets are likely to require different strategies. Because of these reasons we need software where curators can create, compare and evaluate different strategies.

The following use-case will be our guiding example for the remainder of this thesis, which we will use to design, develop and evaluate the software:

> A curator wants to get an understanding which sorting strategy would lead to faster and higher qualitative results. In order to do so he plans to run two nearly identical experiments, only differentiated by sorting method. Each experiment will be represented by an application that should be created by the software. After distributing and running both experiments using a group of crowd annotators he wants be able to compare the results. In order to understand the users actions he wants to monitor these in a quantifiable way that allows for a comparison.

5.4 Measuring the crowd

In order to monitor users actions we need various metrics. A metric is a way of measuring or evaluating a particular phenomenon[28]. These metrics must be observable, directly or indirectly, and be quantifiable in numbers or represented in another way. Examples are the effectiveness (Ex. Being able to find the desired term), efficiency (Ex. The time it took to find that term) or user satisfaction (Ex. How happy a user was with his experience whilst performing this task). A key thing to notice is that these are user metrics, meaning that they are about people and are therefore subject to subjectiveness. When measuring the length of a building, we all agree that it's x meters high, but the happiness of an annotator whilst performing a task is different for each user. By monitoring what a user is doing we hope to gain an insight in where he struggles and how the (implementation of the) process can be improved to better tend his needs. Using the outcome of the metrics we should be able to compare different strategies so an evaluation can be made which strategy works best for a particular goal.

Besides measuring the crowd's actions, we also need to measure the quality of the annotations. The two most common are to use *gold data* (see section 2.2) or by using *majority voting* (counting the number of similar annotations). For example if *windmills* was added ten times and *propellers* only once, *windmills* must be correct by choice of the majority. For most use cases, this is sufficient. However the Iconclass taxonomy has a lot of terms with subtle differences. For example, the most popular annotation on a print could be *Angels* (11G) whilst the more precise and perhaps better options are *angel(s) with sceptre and/or globe* (11G1922), *angel(s) with torch* (11G1923) etc... It could well be that only a minority added one of those annotations. As an addition to ranking the terms on occurrence, we could add a separate *peer review* process where users can rate the tags on their relatedness with regards to the annotatable object.

In doing so, they will add an annotation to an annotation. In table 3.2 we have seen the different operation types of tags. Because we want to add a rating to a tag, options like *Choice*, *Like*, *Score* or *Order* would be fitting for this process. This means that a tag either gets an explicit 1..n rating or an implicit score by a ranking. This extra process should give an insight in which tags are correct and which are not. (For now we will only use gold annotations in the software, but the software should be future proofed to allow for this process in the future.)

5.5 Proposed CH Institute process for testing strategies

Based on the internal process of the RMA RMA(fig. 4.1) and the *Adapted Print Room Online workflow* by Dijkshoorn et al. [21] we've created a process which uses software to test out strategies before a definitive application will be created (Figure 5.3). The yellow blocks are taken from [21] and as such explanations of those can be found in that paper. However the green blocks are added by us to reflect the creation of multiple strategies.

After a set of prints to be annotated has been determined, the curator needs to *determine the annotation goal*. This means that when requesting annotations from the crowd he or she should have a clear goal of the task they want the user to perform. This is important as that goal determines the direction of the strategy.

When a goal has been defined, the software needs to be configured so that it can *Setup an experiment* which fulfils the goal set by the curator. He needs to be able to select a couple of plugins in the software. A unique combination of dataset, taxonomy and input plugins define a strategy. These plugins extend the software with features on an as needed basis and should allow a curator with low technical skills to create an application without much headache.

Of course it is impossible for the software to support all the strategies one can think of from the get go. Whilst the software initially provides some basic plugins, unique strategies will demand unique plugins. In order to extend the software with new features a *developer should be asked to create a plugin*. Based on the needs of the curator the software developer will *create a new plugin* so it can be used to achieve their goal. If it's a generic, non-institute specific, plugin it would be best to share it in a repository so other institutes could benefit from this functionality.

After the setup of the application has been completed and a distribution method has been chosen, a group of workers will *extend the RMA's basic annotation with Crowd Annotations* using the selected strategy. Based on the annotations provided by the users the curator needs to *evaluate* the created strategy by comparing it to another application or baseline so a decision can be made if it's the taken approach is worth pursuing further. Based on this decision the strategy can be refined in another experiment or a completely different one can be created. After a strategy has been deemed to be good enough to be taken into production, the developer should *create a definitive application* which will be used on a large scale. After some quality checks these annotations can then flow back into the RMA's systems, adding additional annotations to the selected prints.

For our use-case the goal has been determined to determine which strategy would lead to faster and better results. We assume that standard input plugins are already present in the software but a developer might be needed to develop some sorting plugins with accompanying metric plugins. After these have been created the curator can setup the experiment, after which the crowd can extend the base annotations. Should the evaluation favour one sorting method we can create an optimised application which can be used on a larger scale or otherwise we need to repeat the experiment with different parameters.

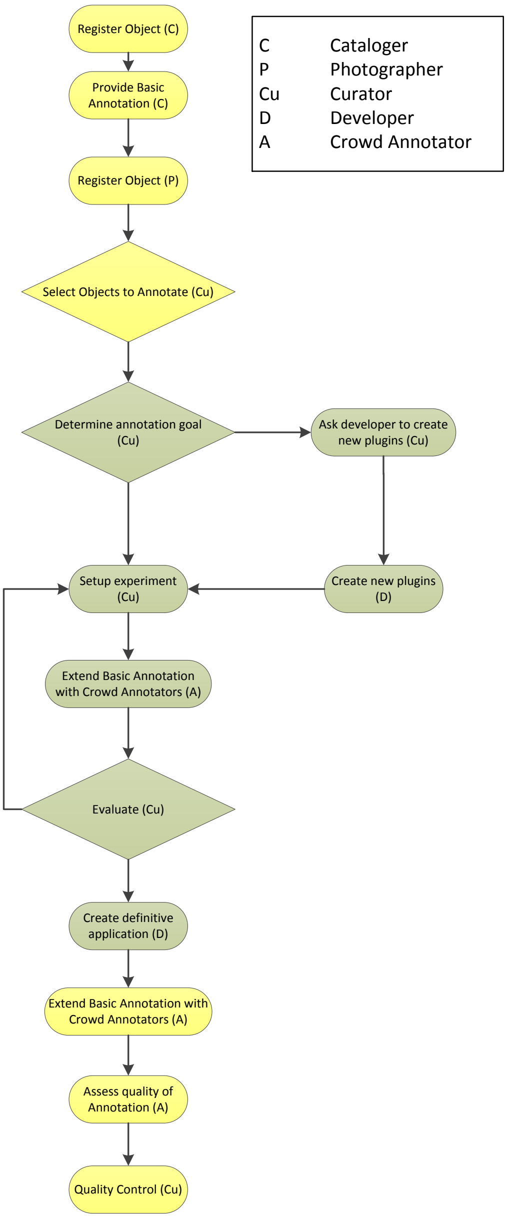

Register Object (C)
Provide Basic Annotation (C)
Register Object (P)
Select Objects to Annotate (Cu)
Determine annotation goal (Cu)
Ask developer to create new plugins (Cu)
Setup experiment (Cu)
Create new plugins (D)
Extend Basic Annotation with Crowd Annotators (A)
Evaluate (Cu)
Create definitive application (D)
Extend Basic Annotation with Crowd Annotators (A)
Assess quality of Annotation (A)
Quality Control (Cu)
C Cataloger
P Photographer
Cu Curator
D Developer
A Crowd Annotator

Figure 5.3: Potential RMA process which includes a software based testing platform

Chapter 6

The Workbench

Based on the previous chapter we have defined a set of requirements for the software, which we will call the workbench. Then we will discuss the features of the application and how it was made. Afterwards we will describe it's competences and to conclude this chapter we will determine if the workbench meets our demands by comparing it to two existing CH annotation applications.

6.1 Requirements

In the first section we will list some requirements based on the user perspective and in the next section we will specify more technical system requirements.

6.1.1 User perspective

CH perspective

The curator employed by the CH institution is the person who will need to setup the experiment. Because he or she is unlikely to feel at home running batch scripts and executing programs from a terminal, we want to make the experience of **creating and setting up an application as easy as possible**. Therefore the steps necessary for configuring an application should be kept to a minimum and easy to understand. The application should then be **easily distributable** among workers and their results should be presented in such a way that it is **easy to understand** how the users performed with regards to the set goals.

Crowd perspective

From the perspective of a crowd annotator, they should be able to **access the application in a simple way** without the need to install any tools. Afterwards they should need a **short amount of time to understand** how the application works, so more time can be spent on the actual task. Finally the application should be capable of **supporting the user** in finding the best annotation for a specific print.

Developer perspective

Because a workbench is a playground support for extra functionalities should be **easy to add**. As a result functionalities can be extended in a short amount of time, keeping the experimentation cycle short. When developing adding functionality the platform should not be the limiting factor of what can and cannot be supported. Therefore the platform should be **as open to configuration as possible**.

6.1.2 Systems perspective

Creating strategies

Since we are unable to implement all the strategies ourselves and a strategy is likely dependent on the needs of a CH institute, the workbench needs to **facilitate the implementation of custom strategies**. Figure 5.2 already illustrates that we have a couple of independent steps and as a result **each step should be implementable by the means of a plugin**. The relations between each element in that figure show that most plugins are dependent on each other and plugins can rely on a specific dataset or taxonomy. Therefore we need a plugin system where **only compatible plugins can be selected** based on their predecessors.

After the experiment has been completed the results should **be compared to a baseline or another strategy with a small variation**[29]. Doing so we can gain an insight in the way the strategy performs and if it meets its goals.

Gathering statistics

In order to compare and evaluate the different strategies we will need various metrics about the users actions. Although we can look at the users output, we would like to know how they have reached this result. When researching the user experience it is common to sit next to the user and ask questions about their actions. Although curators can do this on a smaller scale it this can't be accomplished using the target audience who might be in another county. Therefore, similar to Google's Analytics [1], we would like to add functionality to the workbench which makes it possible to **monitor the actions of the user using various metrics**.

Quality Assurance

It is known that there are malicious users or ones that don't understand the assignment, which as a result will provide wrong answers. In order to determine who these users are we **need to have methods to predict which annotations will likely be of high quality** A popular way of doing so is by using gold data which we've described in section 2.2. Because the workbench should be testbed for testing out different strategies we assume that gold data is always available.

[1] http://www.google.com/analytics/

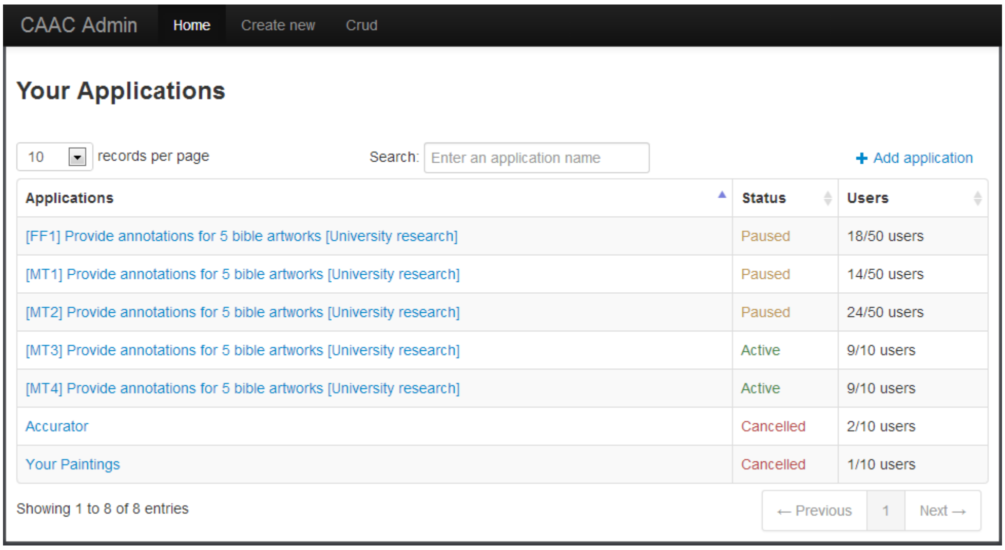
6.2 Features

In this section we will give an overview of the functionalities contained within the current implementation of workbench. In the following sections we will describe the platform and it's underlying architecture in more detail.

6.2.1 Your Applications

The workbench starts with the *Your Applications* page, as seen in figure 6.1. Using the workbench each curator is able to create a so-called application. An application can be used to carry out a specific experiment. This page gives an overview of all the applications a curator has created and gives limited information about them. In this way a curator can see with the glimpse of an eye which applications are currently active and how many users have visited each application. The applications can also be sorted on a certain attribute and, in case the list gets really long, searched based on the application title. By clicking on the application name the *Dashboard* page will be opened, which will be discussed in section 6.2.2.

This page also lets the curator create new applications by clicking on the *+Add Application* button. After clicking on this button the curator will be redirected to a new page where he needs to fill in the title for the application and select a plugin that will act as the interface with the dataset, as the dataset will define the rest of the application. For example it determines which sort plugins can and cannot be used.



Your Applications

10 records per page Search: Enter an application name + Add application

Applications	Status	Users
[FF1] Provide annotations for 5 bible artworks [University research]	Paused	18/50 users
[MT1] Provide annotations for 5 bible artworks [University research]	Paused	14/50 users
[MT2] Provide annotations for 5 bible artworks [University research]	Paused	24/50 users
[MT3] Provide annotations for 5 bible artworks [University research]	Active	9/10 users
[MT4] Provide annotations for 5 bible artworks [University research]	Active	9/10 users
Accurator	Cancelled	2/10 users
Your Paintings	Cancelled	1/10 users

Showing 1 to 8 of 8 entries

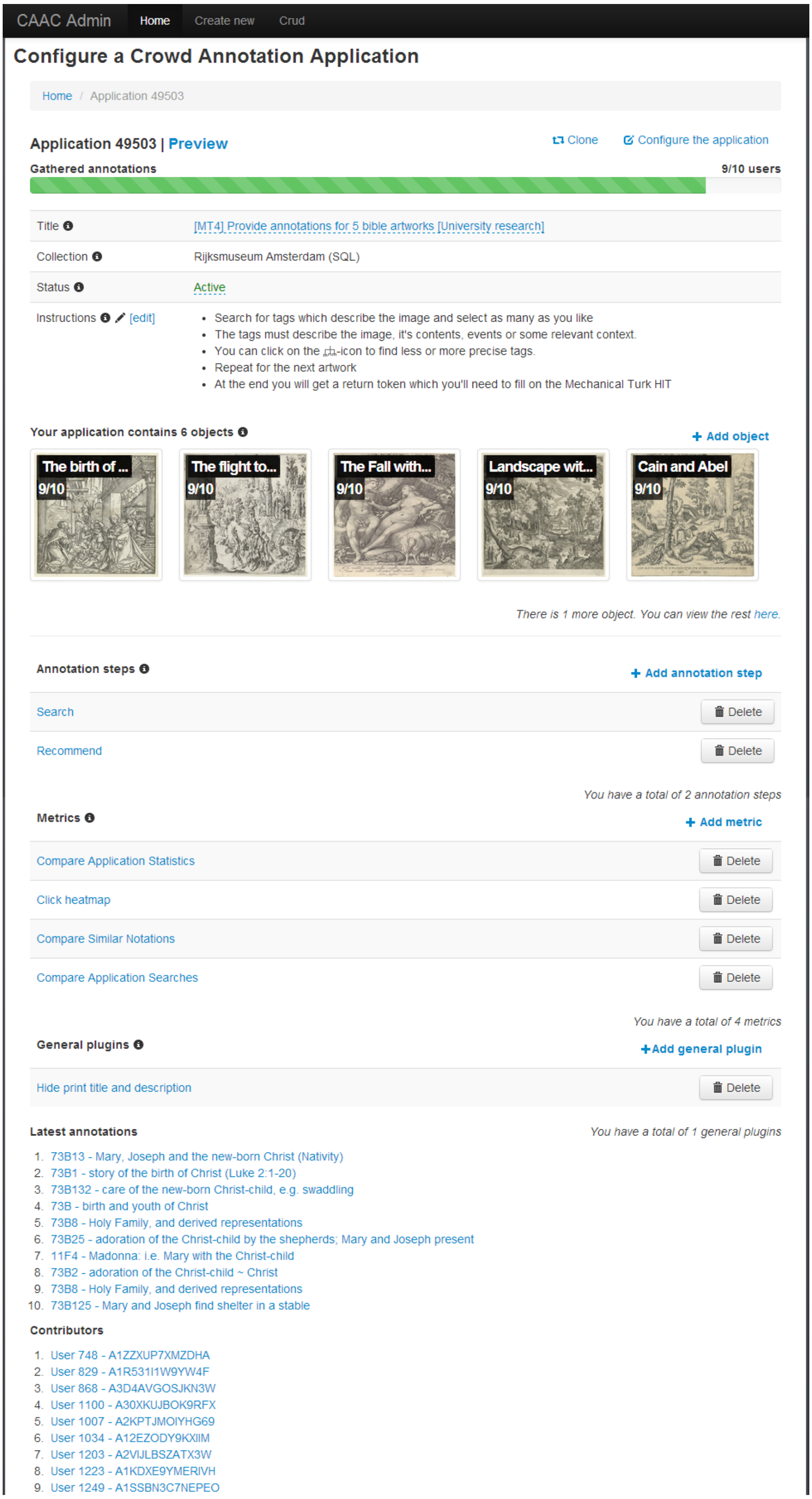
Figure 6.1: Different applications a curator has created

6.2.2 Dashboard

The *Dashboard* page, as shown in figure 6.2 is the main starting point for an application. It tries to give a complete overview and the ability to manage all aspects of the application. Here the curator can …

1. … see a preview of the annotation page the worker will see (see section 6.2.10)
2. … clone an application by pressing the clone button. This will copy everything except the user generated data and the selected plugins of the Annotation Steps (as these are likely to change).
3. … change the configuration (see section 6.2.5).
4. … see the percentage of the experiment that has been completed.
5. … change the application title
6. … change the status. There are four options, `Active`, `Paused`, `Completed` and `Cancelled`. Workers can only see `Active` applications.
7. … define the instructions for a worker. They will see them on the instructions page, after a user accepts the task, and can viewed at any time in the application by pressing the "Reread instructions" button (as can be seen in figure 6.10).
8. … modify and add objects the worker needs to annotate (see section 6.2.3 and 6.2.4)
9. … modify and add Annotation steps (see section 6.2.6)
10. … modify and add metrics (see section 6.2.8)
11. … modify and add general plugins (works similar to metric plugins)
12. … get a quick overview of the latest annotations
13. … get a quick overview of the latest users

The majority of these aspects will be discussed in more depth their respective sections.



Configure a Crowd Annotation Application

Home / Application 49503

Application 49503 | Preview

Gathered annotations **9/10 users**

Title	[MT4] Provide annotations for 5 bible artworks [University research]
Collection	Rijksmuseum Amsterdam (SQL)
Status	Active
Instructions [edit]	• Search for tags which describe the image and select as many as you like • The tags must describe the image, it's contents, events or some relevant context. • You can click on the -icon to find less or more precise tags. • Repeat for the next artwork • At the end you will get a return token which you'll need to fill on the Mechanical Turk HIT

Your application contains 6 objects

There is 1 more object. You can view the rest here.

Annotation steps

Search — Delete

Recommend — Delete

You have a total of 2 annotation steps

Metrics

Compare Application Statistics — Delete

Click heatmap — Delete

Compare Similar Notations — Delete

Compare Application Searches — Delete

You have a total of 4 metrics

General plugins

Hide print title and description — Delete

You have a total of 1 general plugins

Latest annotations

1. 73B13 - Mary, Joseph and the new-born Christ (Nativity)
2. 73B1 - story of the birth of Christ (Luke 2:1-20)
3. 73B132 - care of the new-born Christ-child, e.g. swaddling
4. 73B - birth and youth of Christ
5. 73B8 - Holy Family, and derived representations
6. 73B25 - adoration of the Christ-child by the shepherds; Mary and Joseph present
7. 11F4 - Madonna: i.e. Mary with the Christ-child
8. 73B2 - adoration of the Christ-child ~ Christ
9. 73B8 - Holy Family, and derived representations
10. 73B125 - Mary and Joseph find shelter in a stable

Contributors

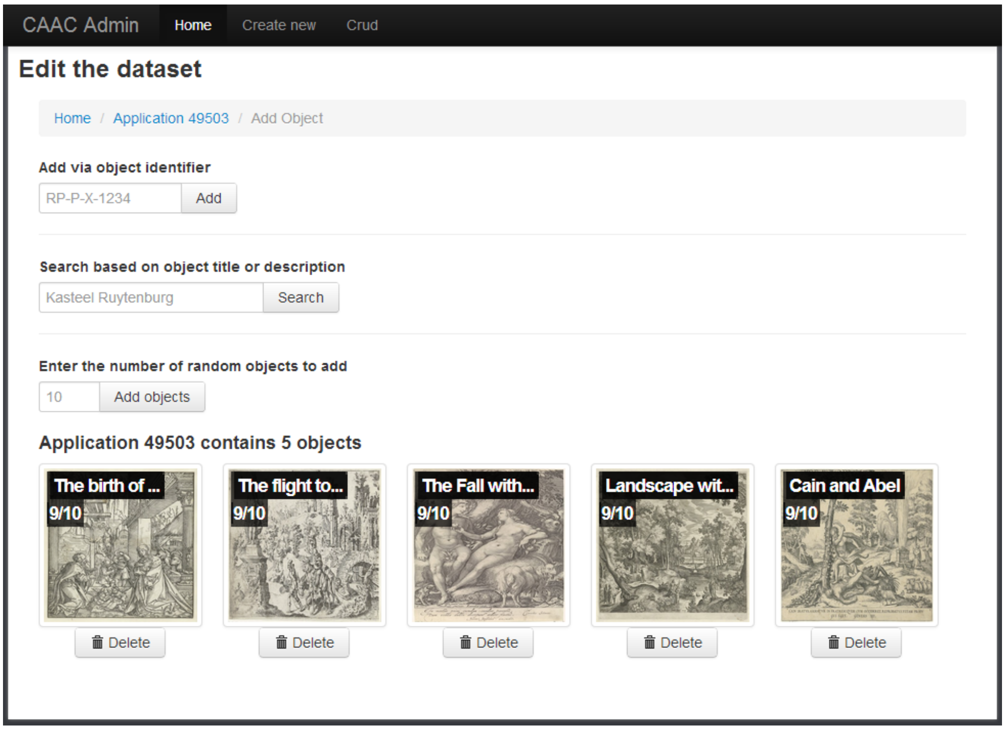
1. User 748 - A1ZZXUP7XMZDHA
2. User 829 - A1R531I1W9YW4F
3. User 868 - A3D4AVGOSJKN3W
4. User 1100 - A30XKUJBOK9RFX
5. User 1007 - A2KPTJMOIYHG69
6. User 1034 - A12EZODY9KXIIM
7. User 1203 - A2VIJLBSZATX3W
8. User 1223 - A1KDXE9YMERIVH
9. User 1249 - A1SSBN3C7NEPEO

Figure 6.2: The dashboard to manage different aspects of an annotation

6.2.3 Add objects

The curator can add objects in three different ways; a) they can use the direct way by adding an object based on object number or database id (they are not the same, but both unique), b) they can search objects based on title or description and c) by randomly selecting x objects from the entire institutes dataset. When the curator submits his search query, a page will be shown with all the results of said query and a selection can be made which objects they want to add to add to the application. Unwanted objects can easily be deleted by pressing the respective button.

Because we intend to deal with different institutes and therefore different datasets, the actual queries will be performed with the use of a dataset plugin. Therefore each database plugin should provide an interface for at least these three options. Plugin creators are of course free to add their own additional search boxes after their plugin has been selected.



CAAC Admin Home Create new Crud

Edit the dataset

Home / Application 49503 / Add Object

Add via object identifier

RP-P-X-1234 Add

Search based on object title or description

Kasteel Ruytenburg Search

Enter the number of random objects to add

10 Add objects

Application 49503 contains 5 objects

The birth of ... 9/10 Delete

The flight to... 9/10 Delete

The Fall with... 9/10 Delete

Landscape wit... 9/10 Delete

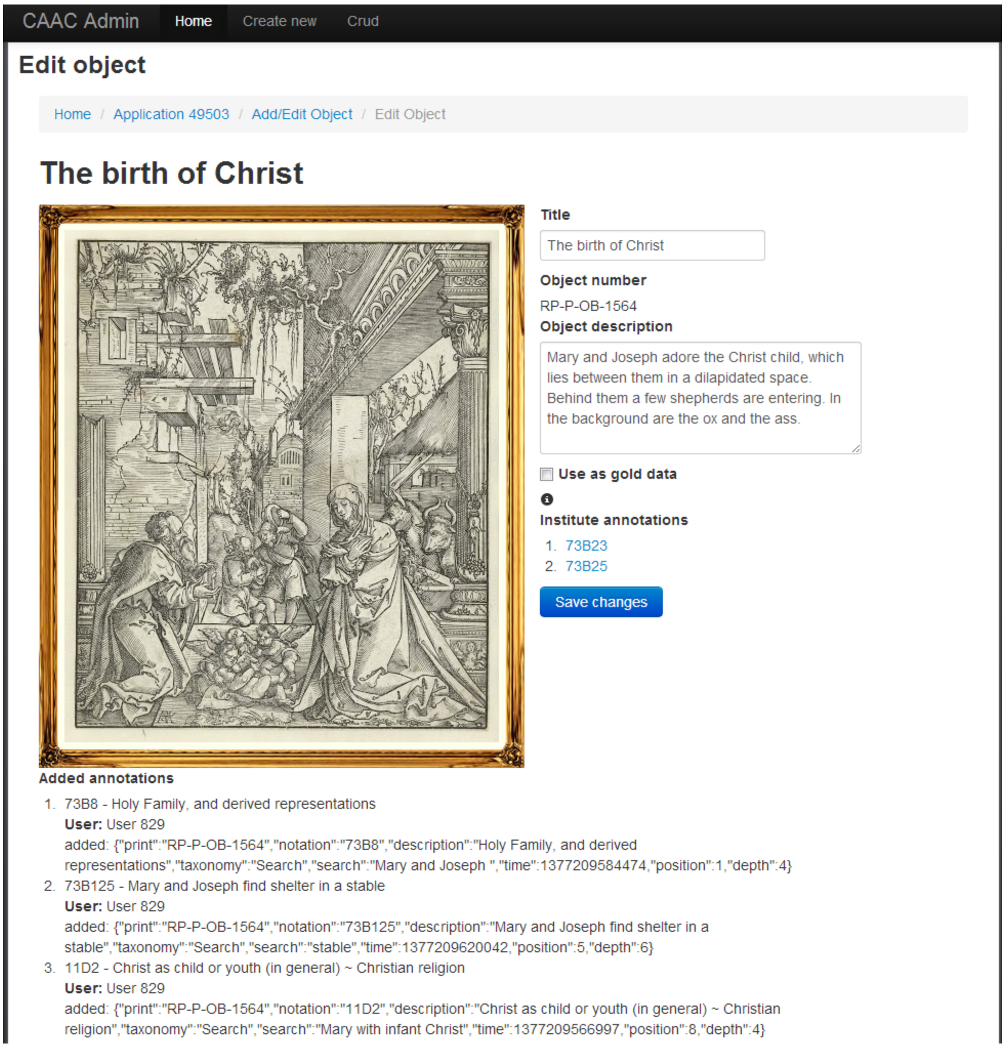
Cain and Abel 9/10 Delete

Figure 6.3: Adding objects to an application

6.2.4 Editing an object

After an object has been added, some attributes can be modified by the curator. For example, Dutch institutes may only have titles and descriptions in Dutch. This might not be convenient since in 2010 80,8% of the workers on Mechanical Turk was either from the United States or India[6]. Therefore the option is provided edit these attributes in order to translate them into English. Another option is to mark certain objects as gold data. This means that a worker must answer (a percentage of) these objects correctly or his result will be rejected and therefore will not be paid. Currently a user will pass the gold criteria if the distance of his annotation is a certain percentage away from the annotations already provided by the institute.

Even though two different applications can use the same print, both are stored individually in the workbench its own database as both can have attributes edited differently by the curator. Finally an overview is given of the work provided by users who have annotated this object.



Edit object

The birth of Christ

Title

The birth of Christ

Object number

RP-P-OB-1564

Object description

Mary and Joseph adore the Christ child, which lies between them in a dilapidated space. Behind them a few shepherds are entering. In the background are the ox and the ass.

□ **Use as gold data**

Institute annotations

1. 73B23
2. 73B25

Added annotations

1. 73B8 - Holy Family, and derived representations
   User: User 829
   added: {"print":"RP-P-OB-1564","notation":"73B8","description":"Holy Family, and derived representations","taxonomy":"Search","search":"Mary and Joseph ","time":1377209584474,"position":1,"depth":4}
2. 73B125 - Mary and Joseph find shelter in a stable
   User: User 829
   added: {"print":"RP-P-OB-1564","notation":"73B125","description":"Mary and Joseph find shelter in a stable","taxonomy":"Search","search":"stable","time":1377209620042,"position":5,"depth":6}
3. 11D2 - Christ as child or youth (in general) ~ Christian religion
   User: User 829
   added: {"print":"RP-P-OB-1564","notation":"11D2","description":"Christ as child or youth (in general) ~ Christian religion","taxonomy":"Search","search":"Mary with infant Christ","time":1377209566997,"position":8,"depth":4}

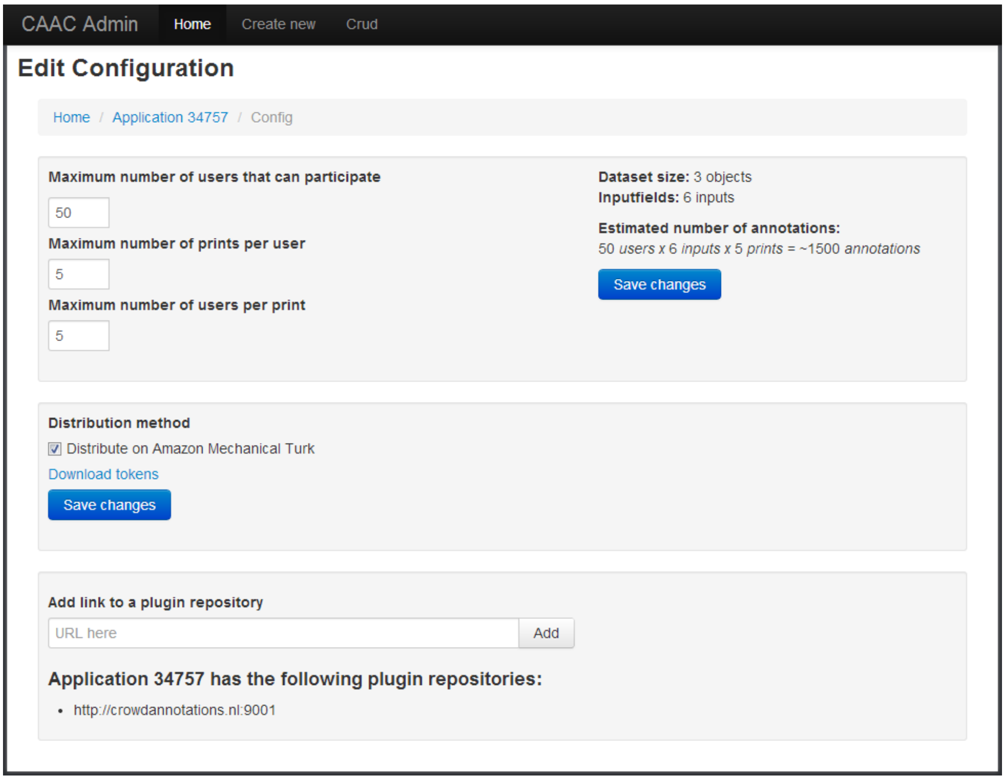
Figure 6.4: Editing an object of the dataset

6.2.5 Config

In the configuration screen general settings about the application can be set. These are the total maximum number of users that can participate in an experiment, the amount of prints per user and the number of users per print. The last two numbers are important in the case a lot of objects are added to the application. For example, if the application contains 100 objects, it isn't enjoyable for a worker to annotate all of them in one go. Therefore a random selection of prints of the max number of objects per user can be made for each worker. Each print will then only be assigned to the maximum number of users, to prevent that a select group of prints will get all the annotations.

As distribution method only Amazon Mechanical Turk is supported at the moment. Currently the user should manually upload a list of tokens to MTurk, but when using the MTurk API it should be possible to automatically list the HITs as soon as the application status is changed to *Active*.

Finally, this screen offers the ability to add plugin repositories: different servers that contain a collection for plugins. Why we need these and more details about the plugin repositories will be given in section 6.5.3.



Edit Configuration

Home / Application 34757 / Config

Maximum number of users that can participate: 50

Maximum number of prints per user: 5

Maximum number of users per print: 5

Dataset size: 3 objects
Inputfields: 6 inputs

Estimated number of annotations:
50 users x 6 inputs x 5 prints = ~1500 *annotations*

Distribution method

☑ Distribute on Amazon Mechanical Turk

Add link to a plugin repository

URL here

Application 34757 has the following plugin repositories:

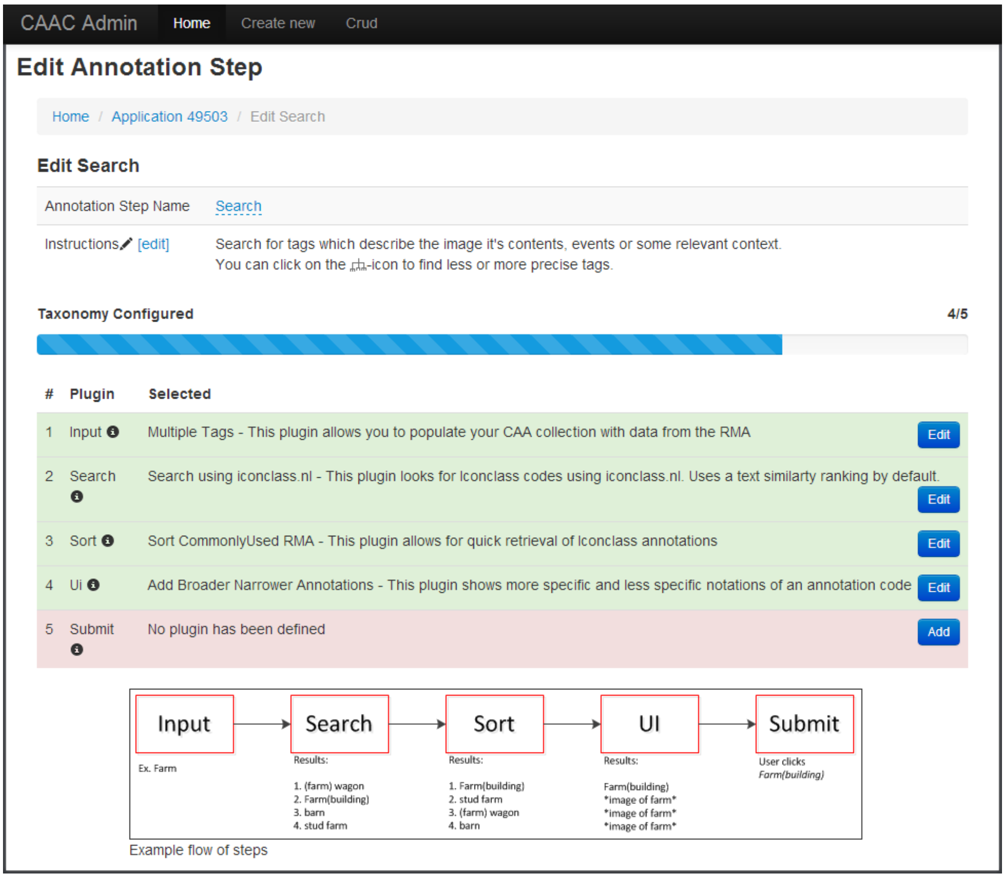
- http://crowdannotations.nl:9001

Figure 6.5: Adding a plugin

6.2.6 Configure Annotation Step

The ability to create Annotation Steps is the main reason we created the workbench. We have given the name Annotation Step to a group of five plugins which comprise a strategy. Here the curator will be able to select one plugin for each step of the process (and more than one in the case of the UI step).

Because no Annotation Step will be the same and will require different actions of a worker, each step should be provided with a title and instructions describing the actions the worker should perform. The progress of successfully setting up a Step has been visualised in two ways; a progress bar and coloured rows denoting which steps have been successfully setup. Each step must be added in sequential order, since the Sorting plugin can be dependent on the Search plugin.



Edit Annotation Step

Home / Application 49503 / Edit Search

Edit Search

Annotation Step Name Search

Instructions [edit] Search for tags which describe the image it's contents, events or some relevant context.
You can click on the -icon to find less or more precise tags.

Taxonomy Configured 4/5

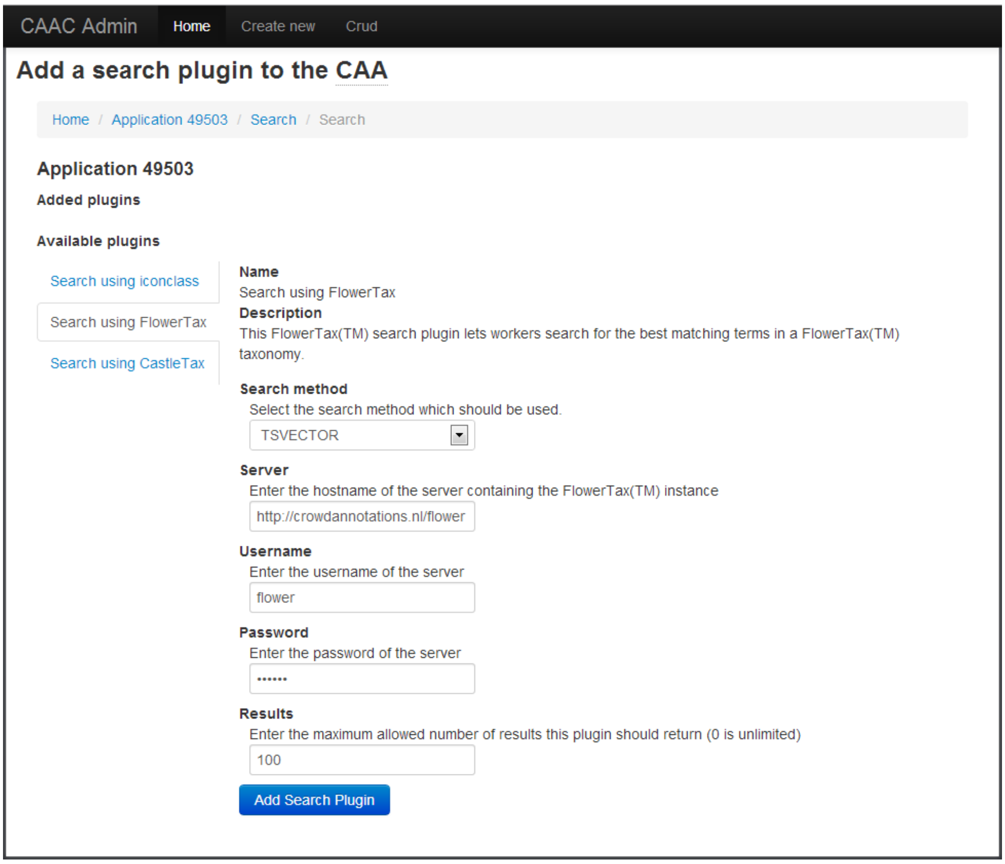
#	Plugin	Selected	
1	Input	Multiple Tags - This plugin allows you to populate your CAA collection with data from the RMA	Edit
2	Search	Search using iconclass.nl - This plugin looks for Iconclass codes using iconclass.nl. Uses a text similarty ranking by default.	Edit
3	Sort	Sort CommonlyUsed RMA - This plugin allows for quick retrieval of Iconclass annotations	Edit
4	Ui	Add Broader Narrower Annotations - This plugin shows more specific and less specific notations of an annotation code	Edit
5	Submit	No plugin has been defined	Add

Example flow of steps

Figure 6.6: Configuring an Annotation Step

6.2.7 Adding plugins

When one wants to add a plugin, all available plugins from all the repositories will be queried. Based on the plugins added in the previous steps these will be filtered to a list of compatible plugins and shown to the user. Each plugin is defined in an XML file (more about this in section 6.5.1 and Appendix C) which contains all the available information about a plugin. The form fields and its options shown in figure 6.7 are also defined in this XML file, complete with the type of field. The form renderer knows when to draw a selection or password field as a result.



Add a search plugin to the CAA

Home / Application 49503 / Search / Search

Application 49503

Added plugins

Available plugins

Name
Search using FlowerTax

Description
This FlowerTax(TM) search plugin lets workers search for the best matching terms in a FlowerTax(TM) taxonomy.

Search method
Select the search method which should be used.
TSVECTOR

Server
Enter the hostname of the server containing the FlowerTax(TM) instance
http://crowdannotations.nl/flower

Username
Enter the username of the server
flower

Password
Enter the password of the server
••••••

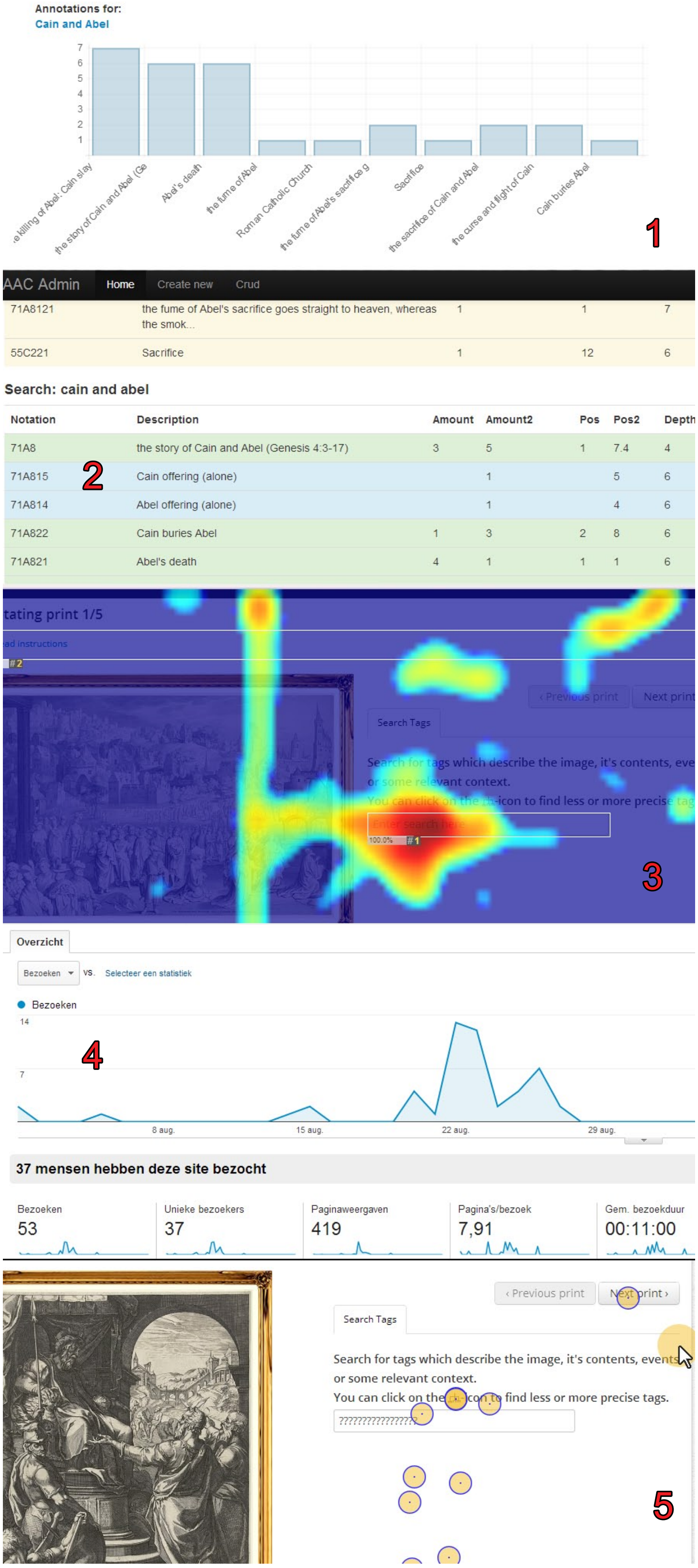
Results
Enter the maximum allowed number of results this plugin should return (0 is unlimited)
100

Figure 6.7: Adding a plugin

6.2.8 Metric Plugins

In order to evaluate new strategies we need the ability to reason about them. The metric plugins allow the curator to record and gain insight about workers actions. In order to do so metric plugins come in two parts, a front- and backend. The frontend plugin gathers information about the user's actions whilst the backend part of the plugin processes these metrics into easily digestible and interpretable content. An example of a backend is given in figure 6.8 image 2. This plugin visualises the annotations that have been added as the result of particular searches.

The Workbench provides an API so that metrics can be stored and retrieved in and from the Workbench its database. However, developers can also opt to use their own external storage. This is for example the case when a plugin for Google Analytics (fig. 6.8 img. 3) has been created, as those results will be stored on Google's server and can be retrieved with the Google Analytics API. By doing so the Workbench aims to provide flexibility to developers.



Annotations for:
Cain and Abel
7
6
5
4
3
2
1
e killing of Abel: Cain slay
the story of Cain and Abel (Ge
Abel's death
the fume of Abel
Roman Catholic Church
the fume of Abel's sacrifice g
Sacrifice
the sacrifice of Cain and Abel
the curse and flight of Cain
Cain buries Abel
1
AAC Admin
Home
Create new
Crud

71A8121	the fume of Abel's sacrifice goes straight to heaven, whereas the smok...	1	1	7
55C221	Sacrifice	1	12	6

Search: cain and abel

Notation	Description	Amount	Amount2	Pos	Pos2	Depth
71A8	the story of Cain and Abel (Genesis 4:3-17)	3	5	1	7.4	4
71A815	Cain offering (alone)		1		5	6
71A814	Abel offering (alone)		1		4	6
71A822	Cain buries Abel	1	3	2	8	6
71A821	Abel's death	4	1	1	1	6

37 mensen hebben deze site bezocht

Bezoeken	Unieke bezoekers	Paginaweergaven	Pagina's/bezoek	Gem. bezoekduur
53	37	419	7,91	00:11:00

Search for tags which describe the image, it's contents, events or some relevant context.
You can click on the -icon to find less or more precise tags.

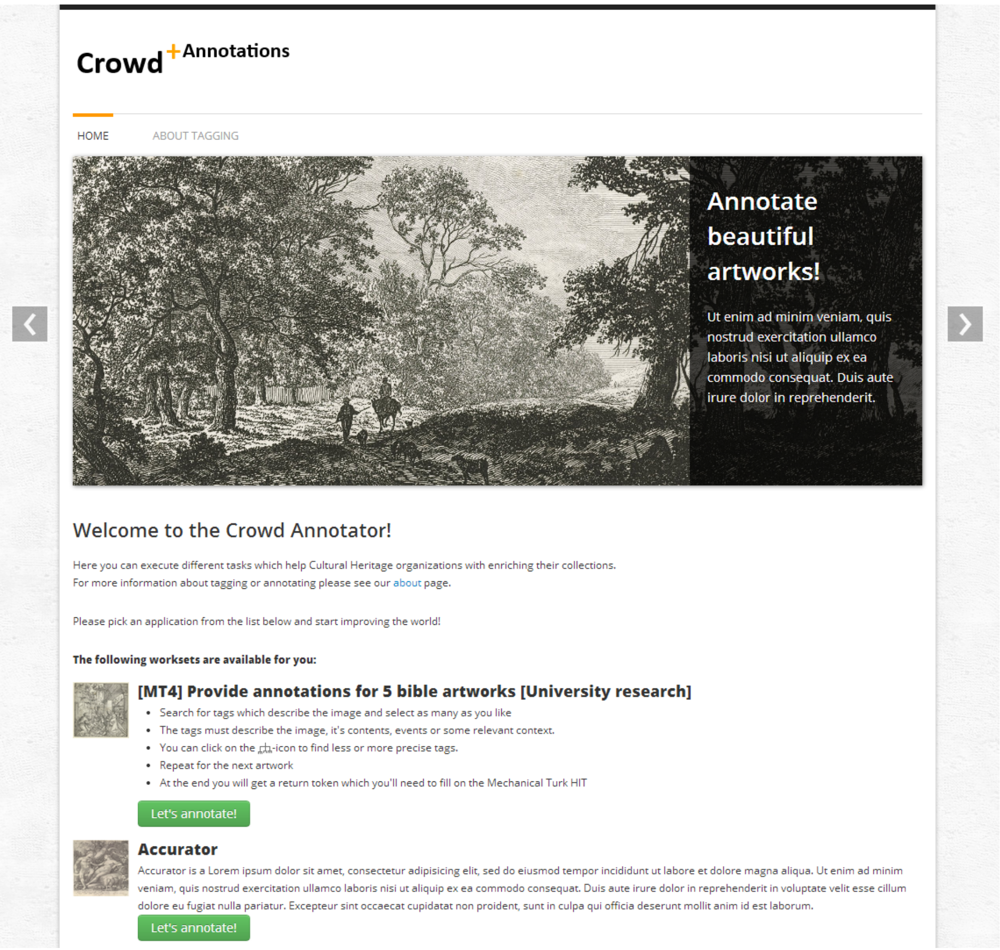
Figure 6.8: Five examples of metric plugins: 1) Provides various statistics, one if which is the amount a term was used 2) Shows which annotations were made after a particular search 3) Shows a heatmap of the mouse actions of all users 4) Common Google Analytics statistics about users and their length (also available in from within the workbench) 5) Shows a video of the mouse actions for one user

6.2.9 User - Available Applications

When a user is not directed to the application via a crowdsourcing website like Mechanical Turk, and visits the workbench platform by visiting the http://crowdannotations.nl url, the user can select one of the applications available on the website (only if an application is marked as publicly available). Unfortunately, if a user performs a task this direct way, he won't be getting any financial compensation. If the user accepts a task on the page shown in figure 6.9 he will be redirected to the instructions page. After reading these he is allowed to use the actual app. This method is useful for distributing tasks among enthusiasts on an institute's mailing-list.

If a worker is redirected to the site from Mechanical Turk, he will skip the landing and introduction pages and can immediately begin at the crowdsourcing task at hand, since the Turk HIT will already contain the instructions.

After the task has been completed, the turker will be given a return token he fills in at the HIT and the casual worker gets a thank you message. A more detailed overview of the MTurk process can be seen in section 7.1.3.



Annotate beautiful artworks!

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit.

Welcome to the Crowd Annotator!

Here you can execute different tasks which help Cultural Heritage organizations with enriching their collections.
For more information about tagging or annotating please see our about page.

Please pick an application from the list below and start improving the world!

The following worksets are available for you:

[MT4] Provide annotations for 5 bible artworks [University research]

- Search for tags which describe the image and select as many as you like
- The tags must describe the image, it's contents, events or some relevant context.
- You can click on the -icon to find less or more precise tags.
- Repeat for the next artwork
- At the end you will get a return token which you'll need to fill on the Mechanical Turk HIT

Accurator

Accurator is a Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

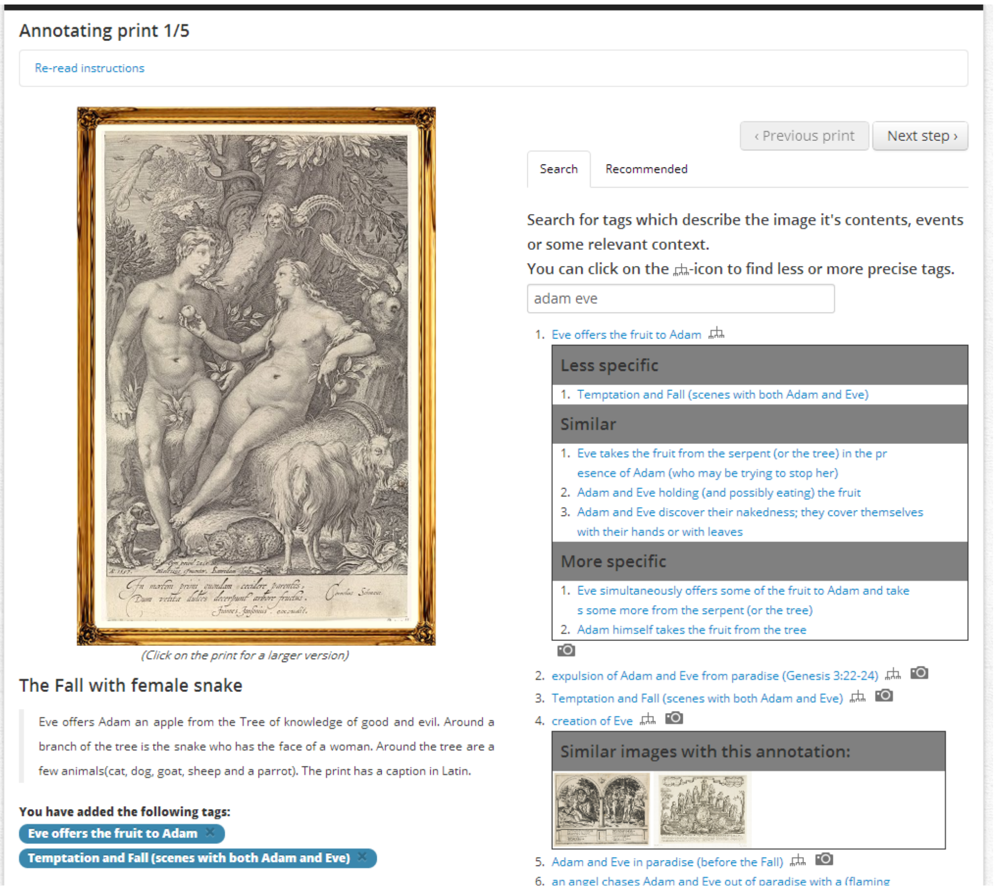
Figure 6.9: List of available applications

6.2.10 User - Executing Experiments

The screen as seen in figure 6.10 is the most important as this is the page that everything revolves around as it executes the implemented strategies. In this case the there are two annotation steps: *Search* and *Recommendation*. In the *Search* step the worker can search for the term he desires to add and is aided by two UI plugins: the ability to select less or more specific terms and the option to view objects with similar annotations. If a suitable term has been determined he can click on it and it will be added to the bottom left (where it can be deleted again if a better annotation is found in perhaps the *Recommendation* step).

After the user is finished with the first step, he can click *Next step* > and the *Recommendation* step will be shown. If there is no next step available the button will change to *Next print* >. Finally the user has the option to reread the instructions and to view the print in full screen.

It is important to note that this is a specific instance and that not all applications will look like this. Therefore in this screen a distinction can be made with functionalities provided by the workbench and ones that are provided via the means of plugins. Generally speaking is the content on the left is provided by the workbench whereas the content on the right is determined by plugins. More will be explained in section 6.5.4.



(Click on the print for a larger version)

The Fall with female snake

Eve offers Adam an apple from the Tree of knowledge of good and evil. Around a branch of the tree is the snake who has the face of a woman. Around the tree are a few animals(cat, dog, goat, sheep and a parrot). The print has a caption in Latin.

You have added the following tags:

Eve offers the fruit to Adam ×

Temptation and Fall (scenes with both Adam and Eve) ×

Search for tags which describe the image it's contents, events or some relevant context.
You can click on the 品-icon to find less or more precise tags.

adam eve

1. Eve offers the fruit to Adam 品

   Less specific

   1. Temptation and Fall (scenes with both Adam and Eve)

   Similar

   1. Eve takes the fruit from the serpent (or the tree) in the presence of Adam (who may be trying to stop her)
   2. Adam and Eve holding (and possibly eating) the fruit
   3. Adam and Eve discover their nakedness; they cover themselves with their hands or with leaves

   More specific

   1. Eve simultaneously offers some of the fruit to Adam and takes some more from the serpent (or the tree)
   2. Adam himself takes the fruit from the tree

   📷
2. expulsion of Adam and Eve from paradise (Genesis 3:22-24) 品 📷
3. Temptation and Fall (scenes with both Adam and Eve) 品 📷
4. creation of Eve 品 📷

   Similar images with this annotation:
5. Adam and Eve in paradise (before the Fall) 品 📷
6. an angel chases Adam and Eve out of paradise with a (flaming

Figure 6.10: Executing experiments

6.3 Platform

When using crowdsourcing you are dealing with a lot of people from all over the world, running various configurations of hardware and operating systems, be it desktop, mobile or tablet devices. Given this information and the fact that the key part of the information presented to the user consists of images and text, the choice for a HTML5 web application was an easy decision. HTML5 apps can be made using various technologies and programming languages, as long as they output HTML, CSS and JS files. It was decided to create this application using the Play Framework which *"makes it easy to build web applications with Java & Scala"* [2]. The Play Framework aids the development process using the Model-View-Controller (MVC) paradigm[3]. Here the application is divided in three sections: 1) The model, where the state of a domain-specific item has been stored, 2) the controller which can manipulate the models and 3) the view which presents the information stored in the model to the user. The Play Framework enforces this paradigm quite strictly. The controllers and models are written in Java whereas the views are written in HTML and the Groovy templating language. Groovy helps avoiding writing duplicate code and allows us to process the models in the view. The Java Persistence API is used to store the Java models in a relational database of your liking (we used PostgreSQL). Besides the MVC paradigm another reason to choose the Play Framework is that it is fully asynchronous and stateless, which makes it easy to deploy on a Cloud Application Platform like Google's AppEngine[4] or Heroku[5] which provide relatively cheap hosting and can scale automatically in case the application gets a lot of visitors.

6.4 Architecture

With regards to the architecture of the workbench in this section we will focus on the models contained in the workbench, as the controllers and views are primarily means to modify and present these.

When you look at figure 6.11 you will see that the `Application` model, which represents an instance of a generated application, acts as the umbrella for the other models. It keeps track of which `Objects` and/or `Annotation Steps` are part of the configuration and defines which collection is used.

Each `Application` should at least contain one `Annotation Step` which represents a single implementation of the annotation process. Each `Annotation Step` must contain one implementation of the five steps outlined in section 5.2.2, namely the `Input`, `Search`, `Sort`, `UI` and `Submit` steps. Each step can be added by the means of a plugin.

Each `Object` can be edited, as seen in section 6.2.4. Here partial data of an object is stored so that a curator can modify it and minimise requests to the CH institutes database since this data isn't prone to change that often.

As seen in figure 6.11 a `Metric` links a `User` to either an `Application`, `Object` or `Annotation`. The reason for these three different types of metrics are the differ-

[2] http://www.playframework.com/

[3] http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

[4] https://appengine.google.com/

[5] http://www.heroku.com

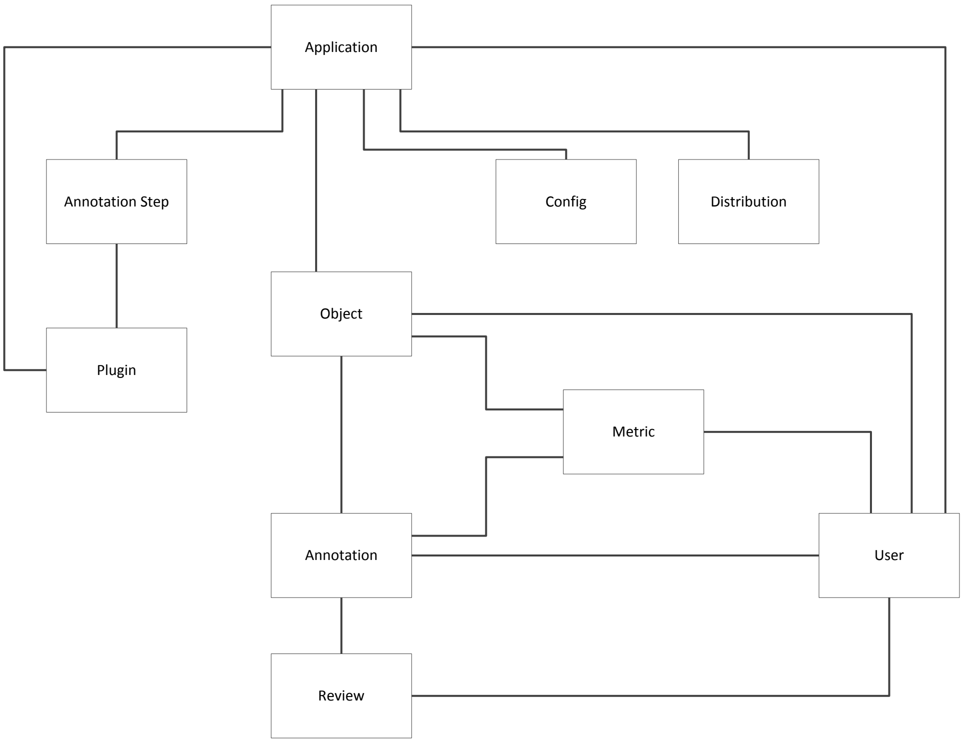


Figure 6.11: Relations between models

ent perspectives. The `User` metric should be used to monitor global user actions, for instance navigation between objects. A `Object` metric for actions taking place whilst visiting a particular object (duration, study time, number of clicks etc.) and finally `Application` metrics for information relevant for a specific annotation (used search term, `Annotation Step` etc.).

A `Metric` consists of a hash-map (`<String,ArrayList<String>>`) where plugins can store key value pairs of information. Although we only allow storage of String values, this doesn't need to be a problem. Since we are working with JavaScript objects can be serialised to JSON strings which can be then stored in the hash-map. As a result of this complex data structures can still be stored in the workbench. And if that isn't enough, plugin makers are free to let their plugin communicate with an external storage of their liking. The same happens when, for example, using Google Analytics.

Because the application can be distributed via crowdsourcing websites like Amazon Mechanical Turk[6] or Crowdflower[7] we don't want each user to go through the process of registering for an account. These people want to annotate as quickly as possible as that way they can make the most money. In order to uniquely identify different transactions we create a `user` based on a sessionId and store his IP, MTurk WorkerId and input token. Thanks to the IP address we can identify where a user is from and a combination of both the IP and WorkerID help prevent the same user from doing the same task twice.

[6] https://www.mturk.com/

[7] http://crowdflower.com/

Each Annotation can be reviewed by multiple users in a possible review process to determine which annotation users deem the most fitting for a print. These reviews are stored in the Review model.

The Distribution model's main purpose is storing a list of entry and success tokens for various crowdsourcing platforms. When a user starts a task on a crowdsourcing platform he gets an unique entry token which gets added as a parameter to the URL the user clicks to enter out website (Ex. http://{website}/{appid}/?token={token}). When the user completes his task on the workbench they get a success token they need to copy to a form on the crowdsourcing platform. This enables us to determine that the task was executed successfully and the user should get paid. At a later stage the Distribution model could also contain a set of distribution plugins, so a curator can distribute the application among crowdsourcing platforms and mailing lists from within the app.

The features of the other models have been described when we discussed the features in section 6.2 and plugins will be discussed in more detail in the following section.

6.5 Plugins

A key part of the workbench are its plugins that allow a curator to quickly create new applications. The Oxford English Dictionary defines a plugin as follows:

Definition 6 (Plugin[8]). *Computing Designating or relating to an item of software or hardware designed to enhance or add a specific feature to a system or application.*

Probably the most well-known example of a plugin is Adobe's Flash Player[9] which adds Flash playing functionalities to a browser.

As we want to make it easy to test multiple strategies, we want to have different implementations for specific features like searching or sorting. Therefore we have developed a custom plugin system that allows a curator to select different implementations of a specific feature with a single click.

6.5.1 Enhancing the workbench with plugins

When trying to develop a plugin system we first looked at existing solutions available for Java. Unfortunately loading plugins in Java isn't so advanced as in other languages. The predominant options were loading plugins using a ClassLoader or the Open Services Gateway Initiative (OSGi)[10] but this is either too limited or needlessly complicated. Plus, since we're dealing with a web application, it didn't allow us to modify the UI on the client side as they don't have any support for defining HTML, CSS or JS resources. After looking at various open-source projects which can be extended with plugins, we settled on an XML Manifest file similar to Eclipse [11] and Apache Maven[12] do, based on our demands. This XML file defines and describes the plugin,

[8] plug-in, Oxford English Dictionary - http://www.oed.com/view/Entry/146024

[9] http://www.adobe.com/products/flashplayer.html

[10] http://www.osgi.org/

[11] http://www.eclipse.org/articles/Article-Plug-in-architecture/plugin_architecture.html

[12] http://maven.apache.org/guides/introduction/introduction-to-the-pom.html

it's resources and configurable options. A full description of the XML's structure can be found in appendix C. The resources referred to by the XML file are HTML, CSS, JS and REST services for requesting and manipulating data.

We opted for web services which return JSON instead of using Java plugins, since we are already working in a web based context. This gives us the flexibility that the web services can be developed in any language as long as they return JSON. An example of such a web service can be an Iconclass search service, which when supplied with a search term returns a list of matching terms.

6.5.2 Plugin types

There are currently eight types of plugins allowed by the workbench, each implementing a specific feature of the workbench. As can be seen in appendix C, each plugin can consist of a HTML, CSS, JS component to extend the front end functionalities (the part that a crowd annotator sees) and the back end (what a curator sees).

Dataset

The Dataset plugin can be used for adding objects from a CH's database to the workbench. Each dataset plugin should support three different methods for accessing prints; by *id*, *search query* and *random* where the first method will return a single object and the latter two a list of objects. Objects returned from those lists can then be approved before they are added to a local copy in the workbench.

Annotaton Step Plugins

The `Input`, `Search`, `Sort`, `UI` and `Submit` plugins combined form an `Annotation Step`. These were the hardest to implement when creating the workbench, for a couple of reasons. First it isn't always a clear case where a step ends. For example, when performing a search, the results often already have an implicit ranking. Another reason is that we need to support a wide variety of options. An `input` does not always need to be a search term, but can also represent an object. And perhaps the most important reason of all; interference. A curator should be able to add multiple `Annotation Steps` and therefore each plugin should not interfere with another. Therefore each annotation step gets its own container to isolate the area's a plugin should modify.

Metric

The `Metric` plugins allow us to gain insight in the user's actions. A metric plugin can use JavaScript to detect click and type events performed by the user and log these events so we can study the user's behaviour. Metrics were explained more in section 6.2.8.

General

General plugins function similar to a metric plugin but with a different purpose. These can be used for general requirements on either the back- or frontend. An example could be adding links and other information resources to the application for an object.

Although this could be done with a `UI` plugin in an `Annotation Step` it is a change that takes effect across multiple `Annotation Steps`.

6.5.3 Plugin repositories

The workbench is a single platform/website where different institutes can create applications and set up experiments. The aim of the workbench is not to be limited to a specific institute. However institute specific plugins like their *Dataset* plugin and sorting plugins that leverage their collection data have no use for other institutes. It is even likely that some institutes don't even want others to use their data and therefore private repositories have been created. This means that each workbench user is free to create and host their own plugin server with their own private plugins. This server can easily be added to the workbench on the configuration screen after which the plugins can be used. General plugins, like a heatmap plugin with no ties to a specific institute, should be hosted in an open Workbench repository for all institutes to use.

An example of a workflow using two repositories and the interaction between plugins is shown in figure 6.12. It is important to note that this interaction isn't optimal and in a production setting the number of requests should be lowered significantly.

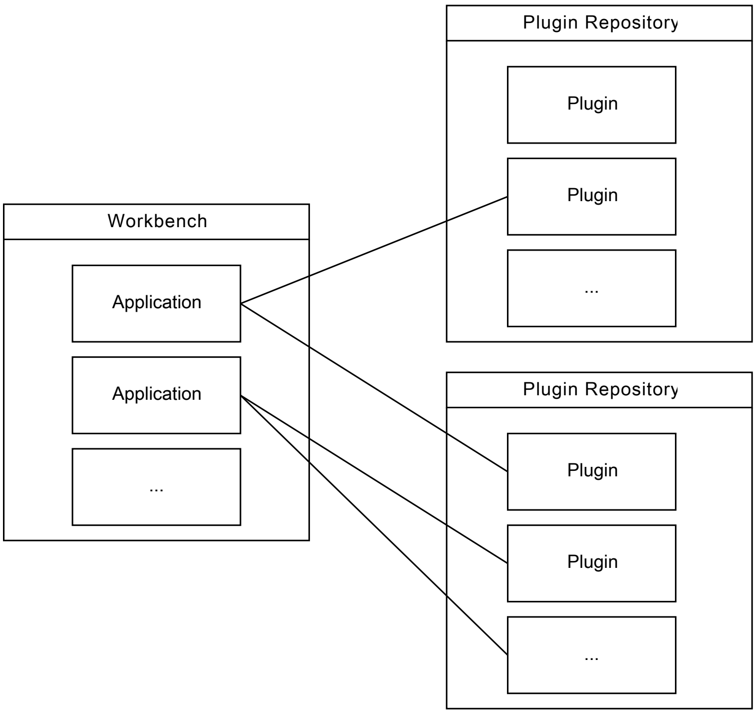


Figure 6.12: Creating applications using two repositories. A full sequence diagram can be found in Appendix B

6.5.4 Application Hooks

Plugins are free to modify and add elements of the HTML's DOM but in order to create structure and a standard platform some core elements were added. The most important here are the `class:annoation_step`, `class:results` and `div:annotations`. The `annotation_step` class is where the combination of Annotation Step plugins will end up. The results should be placed in the `results` class. Because an application can contain multiple Annotation Steps, plugins should be aware that they only modify the correct class corresponding with the Step. This is achieved by adding a unique identifier for every Annotation Step to each class. Finally each plugin should make sure that the results will be added to the `annotations` div.

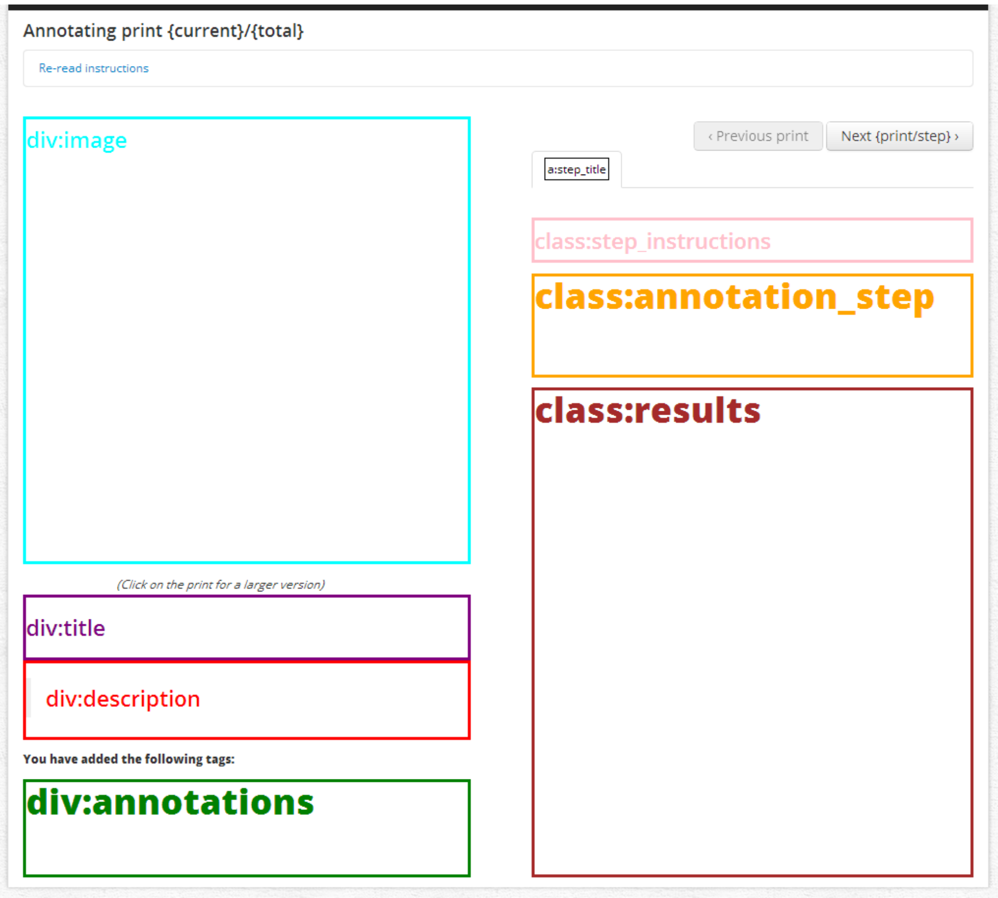
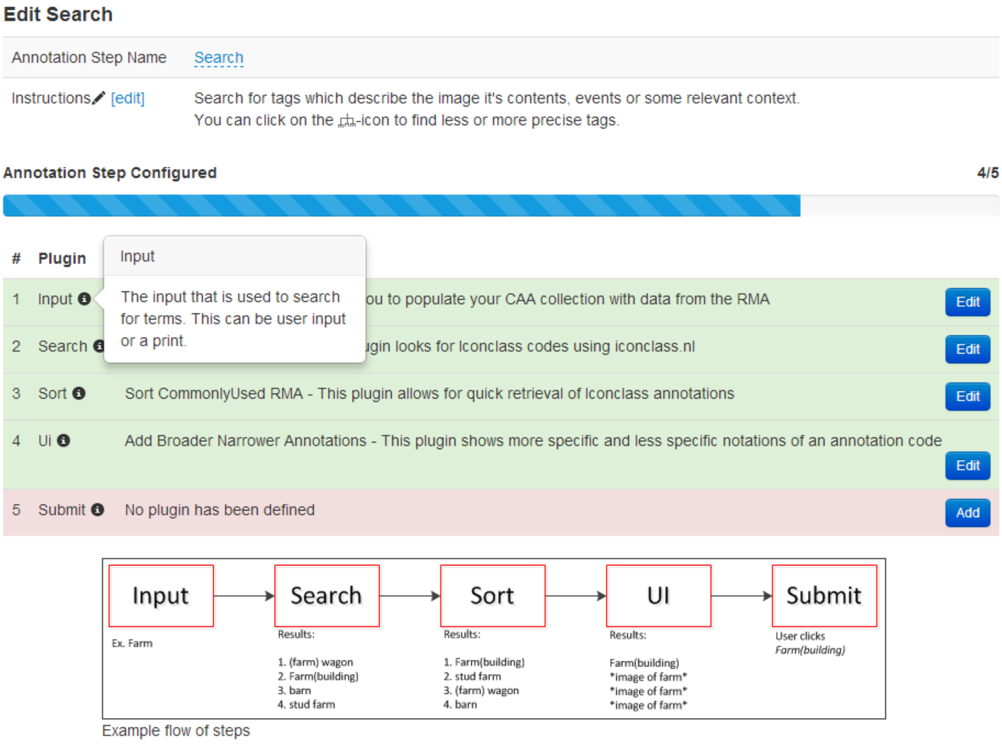


Figure 6.13: Sequence diagram of interactions between plugins and two repositories

6.6 Competences

6.6.1 Usability

The workbench should be easy to use for a curator and we've tried to achieve this in various ways. When developing the workbench we kept a flat navigation structure, meaning that all the key aspects are viewable from the main, or often called dashboard,



Edit Search

Annotation Step Name Search

Instructions [edit] Search for tags which describe the image it's contents, events or some relevant context.
You can click on the -icon to find less or more precise tags.

Annotation Step Configured **4/5**

Input

The input that is used to search for terms. This can be user input or a print.

#	Plugin		
1	Input	ou to populate your CAA collection with data from the RMA	Edit
2	Search	gin looks for Iconclass codes using iconclass.nl	Edit
3	Sort	Sort CommonlyUsed RMA - This plugin allows for quick retrieval of Iconclass annotations	Edit
4	Ui	Add Broader Narrower Annotations - This plugin shows more specific and less specific notations of an annotation code	Edit
5	Submit	No plugin has been defined	Add

Example flow of steps

Figure 6.14: Three methods to aid a curator

page of an application. Each aspect is then no deeper than two levels, lowering the navigational complexity.

When new concepts or application specific terminology was introduced we tried hard to help a curator with giving them an understanding of what something means and/or is supposed to do. As you can see in figure 6.14 when setting up a `Annotation Step` the curator is aided in three different methods. 1) Behind each step of selecting a plugin the curator can click on the ℹ-icon which will open a pop-up with an explanation of what this step represents. 2) At the bottom an example flow is given to provide the curator with a visual representation of the results of each step. 3) Finally he gets visual feedback for the progress of the configuration in two ways. The progress bar denotes the percentage completed, similar to the green coloured steps stating that that part has been successfully setup.

6.6.2 Extensibility - Workbench Web API

When creating plugins it is convenient if there is an interface with the data models. Therefore we have created a pragmatic REST API based on the best practises in [30]. Pragmatic meaning it's designed with a developer in mind instead of strictly following the REST style defined by R.T. Fielding [13]. The API follows the CRUD (Create-Read-Update-Delete) principle where a specific HTTP request stands for a different action for a resource URL, as can be seen in table 6.1. This example shows how to manipulate

[13] http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

metrics for a specific user, but other resources are available as well. Our goal was to make most of the resources mentioned in section 6.4 available to developers in order to, again, be as open as possible.

When a developer is only interested in a subset of the metrics of a particular user, parameters can be added in order to retrieve a smaller result; for example: `GET` <application>/metrics?fields=timein,timeout,searchquery&limit=10&offset=50. Timein, timeout and searchquery are examples of fields we logged for each user.

Request	CRUD	Action
`POST`	Create	Add a new metric to `<application>` for `<user>`
`GET`	Read	Get all metrics of `<application>` for `<user>`
`PUT`	Update	Bulk update all metrics of `<application>` for `<user>`
`DELETE`	Delete	Delete all metrics of `<application>` for `<user>`

Table 6.1: An example of the Web API for /<application>/metrics/<user>

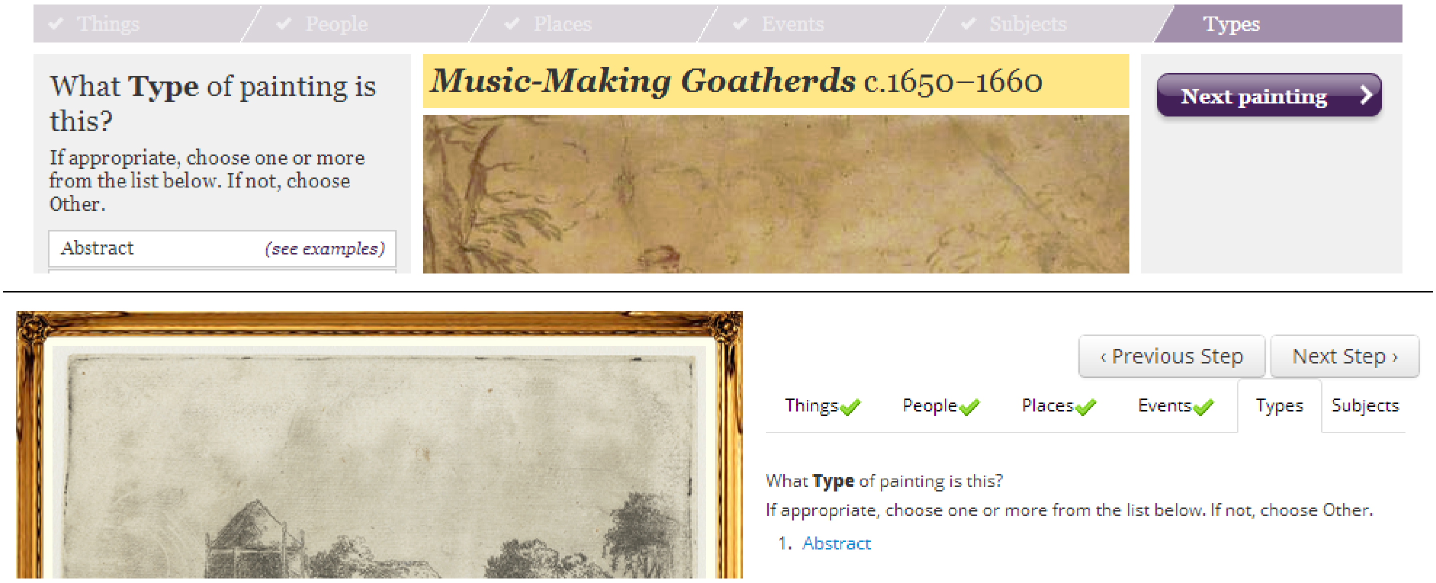
6.7 Existing systems compared to generated applications

In order to determine how capable the workbench is when compared to existing systems we decided to do a comparison with two of the applications we have seen in Chapter 3. We have chosen to replicate Accurator and the Your Paintings Tagger as they are both set in a Cultural Heritage setting and therefore come the closest to the context the workbench has been designed for.

Your Paintings Tagger

Features The Your Paintings Tagger lets users tag in different categories. Users need to add tags in the categories *Things, People, Places, Events, Subjects* and *Types*. The first four categories are based on the entry of free text whereas in the last two the user needs to select from a list of predefined tags. In -case that a user types in the free text field, recommendations are being made. These recommendations are however the same for each category; this means that the word *’hat’* will still be recommended if the user wants to add an event.

Differences When we try to replicate the first four steps in the workbench, there is one major difference; because we require a user to select terms from a taxonomy the perspective has changed from an application where a user freely types a tag to a perspective where a user *searches* for a tag. Aside from this shift of perspective we have no problem of replicating the functionality of Your Paintings. The four free text steps can be setup by creating four Annotation Steps (depicted in section 6.2.6) with a free input plugin.
For generating the content/results we use a search plugin which searches in a large taxonomy containing a mixture of things, people and events, but because each Annotation Step can use its own individual search method, we can do one better and use a People taxonomy which only returns names of persons as a result.



What **Type** of painting is this?

If appropriate, choose one or more from the list below. If not, choose Other.

Abstract (see examples)

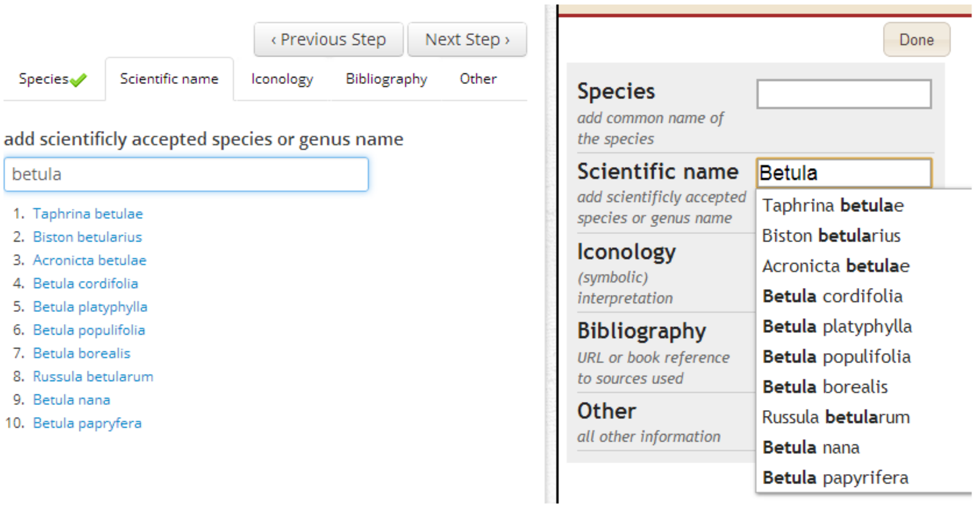
Music-Making Goatherds c.1650–1660

What **Type** of painting is this?

If appropriate, choose one or more from the list below. If not, choose Other.

1. Abstract

Figure 6.15: The Your Paintings Tagger *(top)* and how it is implemented by the workbench *(bottom)*



add scientificly accepted species or genus name

betula

1. Taphrina betulae
2. Biston betularius
3. Acronicta betulae
4. Betula cordifolia
5. Betula platyphylla
6. Betula populifolia
7. Betula borealis
8. Russula betularum
9. Betula nana
10. Betula papryfera

Species — add common name of the species

Scientific name — add scientificly accepted species or genus name: Betula

Iconology — (symbolic) interpretation

Bibliography — URL or book reference to sources used

Other — all other information

Taphrina **betula**e
Biston **betula**rius
Acronicta **betula**e
Betula cordifolia
Betula platyphylla
Betula populifolia
Betula borealis
Russula **betula**rum
Betula nana
Betula papyrifera

Figure 6.16: Accurator *(right)* and how it is implemented by the workbench *(left)*

For the final two categories of Your Paintings, Subjects and Types, where a user needs to select an element from a predefined list we can select an Input plugin which takes no input combined with a search plugin that always returns a predefined list. All six steps then can use no sorting, no UI enhancements and Clicks as submit method.

Accurator

Features Accurator has five input fields, all displayed on the same page, and makes a distinction between two annotation goals. The first three are used to describe the print whereas the other two are used to write down the sources used to determine the answers for the first three. The system uses individual recommendation methods for the first three whereas the "source fields" don't since it's impossible to recommend URLs or books.

Finally Accurator actively recommends prints to annotate to users, based on their preferences and user profile.

Differences As mentioned in the previous section, it is no problem for the workbench to use individual taxonomies for different input fields. However, we aren't able to display all input fields on the same screen as a result of design decisions to ensure the independent working of each plugin and the goal to support a wide array of different plugin types.

The reason is that, by default, the workbench isolates each Annotation Step in its own tab, similar to Your Paintings. The tabs also provide a clear separation for a user between Annotation Steps, which enhances the user experience; especially when dealing with static lists of (recommended) items. When dealing with a static list of items, we soon realised we would encounter some usability issues. For instance: if an application has 2 input fields and 2 static lists all on the same screen, a user would be presented with four different lists of results. This would clutter the screen and makes it harder to spot which list has changed after a new search. We wanted to prevent this behaviour from happening and therefore made the decision to by, default, force a tabbed structure. This structure can of course be altered by making a UI plugin.

Another restriction is that Accurator allows a user to add URL's or book titles in order to add the resource where they found information. Unfortunately, to my knowledge, there are no taxonomies which contain all the URLs in the world. Therefore this kind of input field is hard to represent in the 'search-in-a-taxonomy'-perspective we want to enforce in the Workbench. We could replicate this by creating a search plugin which simply returns the results of the input plugin and use an auto-submit plugin to mimic the workings of a regular free text input field, but this kind of defeats the taxonomy oriented nature of the Workbench.

Last, with regards to the distribution of prints, this is something we cannot do because a user would need to have a profile for this. Because we distribute our tasks on MTurk an account would complicate matters, but could be added for returning users. A similar system is being used by Crowdflower which also distributes tasks on MTurk, but let returning users log in to earn CF Bonus points.

Conclusion

The workbench is able to replicate the behaviour of two representative Cultural Heritage initiatives with regards to the most important task of making annotations. However there are some limitations with regards to the user flow. The main cause is that we needed isolate plugins to avoid having a conflict between each other. This will not be a factor should an application be created from scratch, without the workbench, with a clear single purpose. Another aspect is that we have chosen for an uncommon approach where a user must select terms from a taxonomy, to ensure our quality demands, whereas in the compared initiatives terms can be entered freely. Because the workbench was created with taxonomies in mind, free text annotations become more cumbersome than usual.

Chapter 7

Evaluation of Strategies

For the evaluation of our workbench software we want to focus on whether or not we are able to gain an insight in the actions of users and compare applications. We decided to focus on this aspect rather than to evaluate if the workbench is capable in being used or found useful by curators and developers. Because if we cannot compare applications; there is no point in using the workbench. In sections 7.1 till 7.4 we will determine if we can compare two applications and understand user actions while in section 7.5 we discuss the workbench and give some pointers on how a complete evaluation involving the curators and developers can be completed.

For the evaluation of the capabilities of the workbench we will use the use case as described in section 5.3 as our guiding example. We are aware that this is a single instance and the results might differ in others but it should give a sufficient idea if the course taken is correct.

7.1 Evaluation Setup

For the evaluation two nearly identical applications were created. Each application uses the same five prints from the RMA dataset, displayed in the same order and one Annotation Step that uses the Iconclass taxonomy. The first application uses a text-similarity based sorting strategy which we will abbreviate to **STS** for **S**ort **T**ext **S**imilarity. The second strategy sorts the terms based on how often each term was used by a professional which gets abbreviated to **SCU** after **S**ort **C**ommonly **U**sed. STS should favour terms which resemble the query the user has entered whereas SCU terms which professionals already deemed fit for other prints (and match the query). The reason we've chosen for a smaller dataset of five prints is that we want as many impressions and searches for each print as possible. In order to make a good comparison for our sorting method, we need comparable searches and therefore we deemed it more useful to get 10 impressions for 1 image, rather than 1 impression for 10 images.

7.1.1 Plugins

As described in chapter 6 it is sometimes hard to separate functionalities in a separate search and sorting plugin, as often search results are already returned with some sort of order. One reason for this is that often when searching you only want to return the top `k` relevant results. In order to obtain this relevancy the results need to be ranked

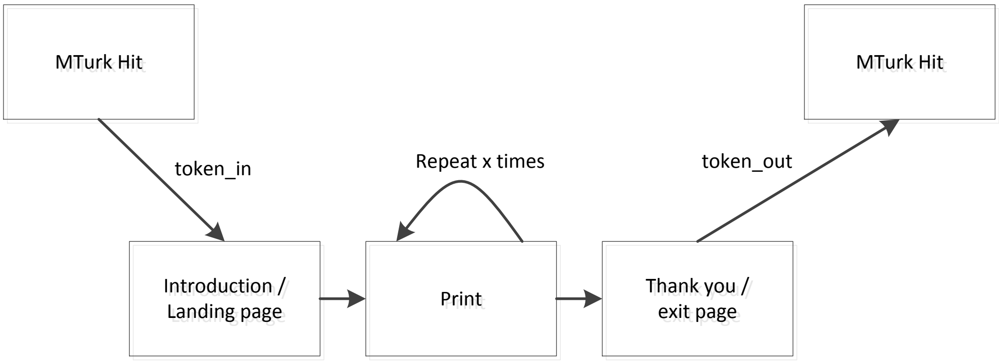


Figure 7.1: Workflow of a MTurk HIT using a token

to determine which are the top `k` results that should be returned. When searching for Iconclass terms we use a plugin which uses Apache's Solr's Full-Text search platform. Solr is a *"popular, blazing fast open source enterprise search platform"*[1] which is a proven approach for searching and sorting based on full-text search. Due to their already advanced sorting algorithms for STS the workbench has been configured retain the sorting of Solr and whilst SCU performs an extra sort.

7.1.2 Metrics

In order to compare the two sorting methods a we chose a couple of plugins which will log all the searches, along with the results, time spent on each print, the navigation between each print, the number of times the user viewed a full screen view of the print, deleted an annotation and reread the instructions. These plugins were selected to get an impression of how difficult an user found the annotation process. For example more searches would indicate that they had problems finding the term they wanted and navigation back and forth can indicate that they have learned something at a next print they want to apply in a previous one.

Because we are comparing two plugins which only differ in the used sorting method we created a plugin which logs the position of a term when it was clicked. We do so to determine the impact of the position of the term and compare which sorting method would provide the best result with regards to showing gold annotations.

7.1.3 Mechanical Turk Setup

After the two applications have been created in the workbench, they need to distributed among workers. The chosen distribution method is Amazon's Mechanical Turk because it is a proven platform for crowdsourcing and provides a large population skilled of workers. Furthermore we can specify that we only want Americans as they usually have a good knowledge of the religious objects in our dataset. A worker usually performs their HIT's (Human Intelligence Tasks) on the MTurk platform itself, but for a curator it is easier to setup an experiment using the workbench and there we can more elaborately monitor a user's behaviour. Therefore the HIT only contains a link

[1] http://lucene.apache.org/solr/

to our website, a field for a return token and a field for any comments or remarks the worker might have. The return token will be given at the end of the experiment at our website, to confirm a user has completed the entire task. A graphical representation of this process can be found in figure 7.1. Tasks on MTurk are usually short, so it is common for a user to tag one object and get rewarded. If the user likes the assignment, he can simply accept another HIT. However, because we already ask extra time of a worker to go to an external website, we deemed it more efficient to annotate multiple objects in a row. In order to prevent a user to perform the same task twice the workbench contains IP and Worker ID checks to prevent a user from doing so. Because we use two different workbench applications, we use an approach similar to A/B-testing [29] where the first user gets redirected to application A, the second to B, the third to A etc. Instead of first handing out assignments for application A and afterwards application B, we decided to interleave the tasks to prevent time related differences with, for example, workers in the East Coast vs West Coast of the USA.

7.2 Pilot study

In order to assess the correct values for the MTurk variables and test the correct functioning of the application we performed a pilot study. As a result of the feedback received during the pilot we found that the workers took more time than expected which resulted in a low hourly wage. Therefore we decided to increase the reward from $0.25 to $0.35. We also noticed a small bug which only occurred in the outdated Internet Explorer versions 7 & 8 which we fixed.

7.3 Evaluation study

The actual evaluation study was performed on Amazon's MTurk using 20 workers, 10 for each application, using a set of 5 prints. Each user was asked to annotate every print. Again we opted for a smaller dataset we would get enough annotations for each print with a small amount of workers.

At the end of the experiment we had 11 valid results on SCU (1 extra because apparently 1 worker forgot to fill in his token on the MTurk website) and 9 valid results using STS. Application STS had 1 rejected user since he didn't add any annotations at all. His hit was republished, but was still open after a week, after which we decided to terminate our experiment. Therefore we decided to delete the extra worker from Application A and another randomly selected one to get an equal 9 vs 9 users for each application.

7.4 Evaluation Results

In this section we will first compare both generated applications on a general level, no specialised plugins required, after which we will try to get a deeper insight with help from two specialised metric plugin we developed.

7.4.1 General statistics

Figure 7.2 shows the time needed for a user to annotate five prints. Users on SCU spend less time doing so when compared to STS. On average the SCU annotators spend 20 seconds less time on the overall process. However, when we look at the time spent on each print, we see that for the first two prints SCU takes longer to annotate than STS. Three possible explanations are: 1) The first two prints might be harder to annotate using SCU 2) Users get more experienced and become more efficient looking for the correct annotation 3) Users become tired and want to finish the task as soon as possible. Based on the limited data gathered we are unable to identify what the correct explanation might be.

Generally speaking we have seen the effect of users becoming 'tired' as both applications contained a few uncompleted/abandoned tasks (we didn't include those in our analysis of the results). Some of those quit after the first print, likely surprised they needed to annotate multiple images (even though it was stated in the MTurk description). However we would expect that the learning and fatigue effects should likely be the same for both applications and thus doesn't explain why SCU performed faster in the last three prints.

What might explain the time difference is that when looking at the searches being done by the users we see that the number of searches goes down (fig. 7.4). This is similar for both applications, but the number of SCU searches drop more drastically. However, when we look at the number of annotations (fig. 7.5), these are a bit steadier. For example, for print 4 the number of searches with SCU is lower as for STS, but the number of annotations is higher. This is clearly visible in figure 7.6 where we look at how many annotations are added after a search. This is perfectly possible with the workbench, as we switched to a search perspective. Similar to what you do on Google where you visit a couple of sites that match your query, people add a couple of annotations that match theirs. As a result of this the number of annotations per minute is also higher (fig. 7.7).

Previous measures do not regard annotation quality; if a method is faster it does not mean that it is better. We compared the crowd annotations with the gold annotations provided by the RMA's professionals. We will do this on two aspects: the distance between the correct annotations to the annotations provided by the users and the number of times users actually added the same gold annotation.

When counting the number times gold annotations were added, it is clear from figure 7.8 that SCU was more successful. From the 16 possible gold annotations the users of SCU added 14 out of 16 and STS scored much lower with 9 out of 16. In total the users of SCU added 56 gold annotations and STS 40. However this might not be completely caused by the different sorting order. For example three gold annotations that were added by the SCU users are of animals portrayed on a particular print. These annotations were added by two users after searching on the term *animals*. When investigating why these weren't added by STS users it became clear that none of the users searched on that or any animal related term for that print. Another reason is that SCU sorts on terms which are used by the professionals. So this introduces a bias towards terms that professionals use. Ultimately this is what we want, but for this

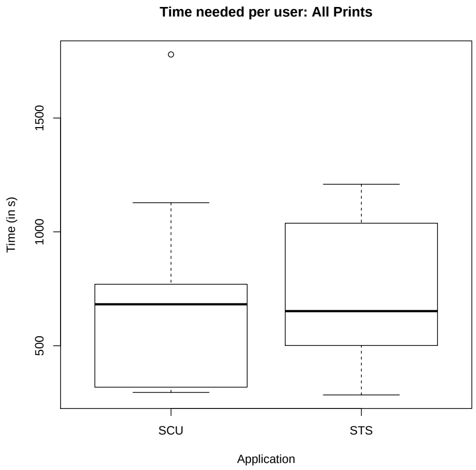


Figure 7.2: Time needed for a a user to annotate all prints

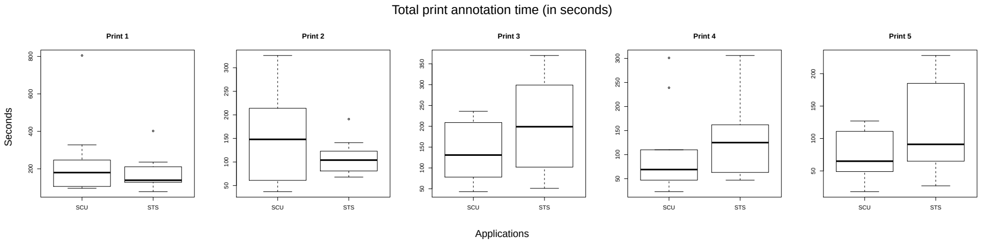


Figure 7.3: Time needed for a user to annotate a print

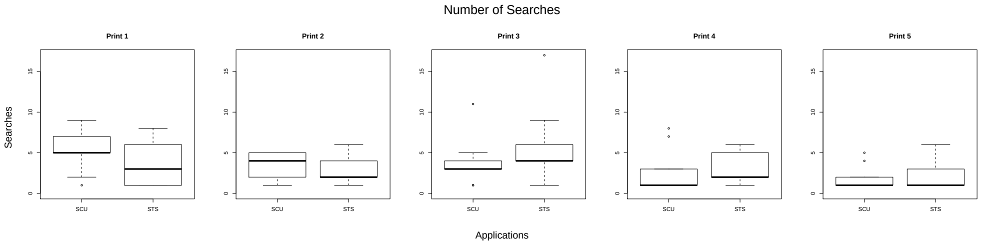


Figure 7.4: Number of searches a user used for a print

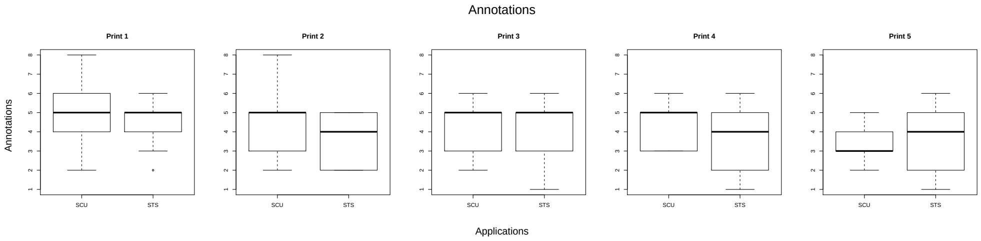


Figure 7.5: Annotations for a user per print

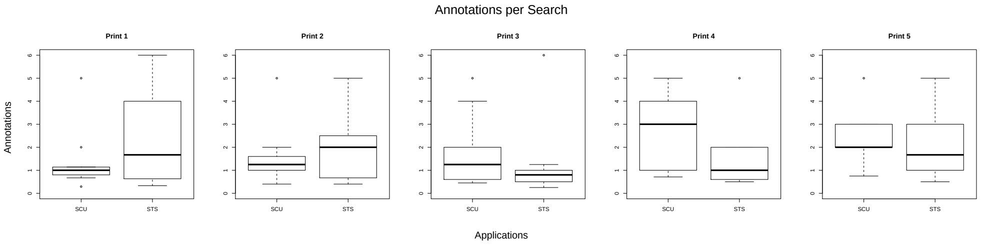


Figure 7.6: Number of annotations added after a search query

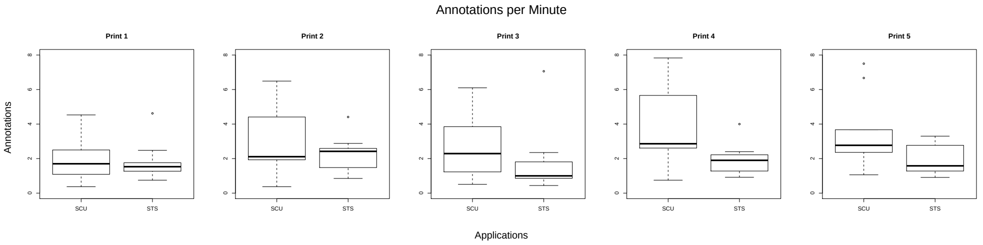


Figure 7.7: Number of annotations per minute

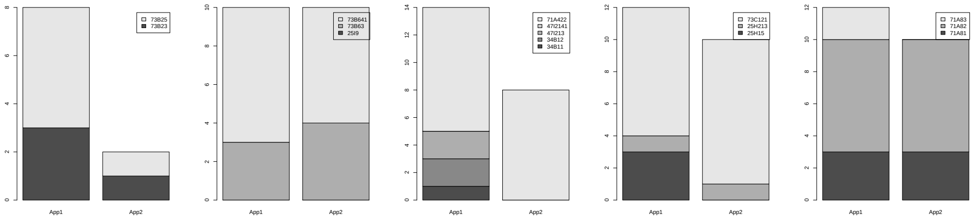


Figure 7.8: Number of annotations per print that are similar to the gold annotations

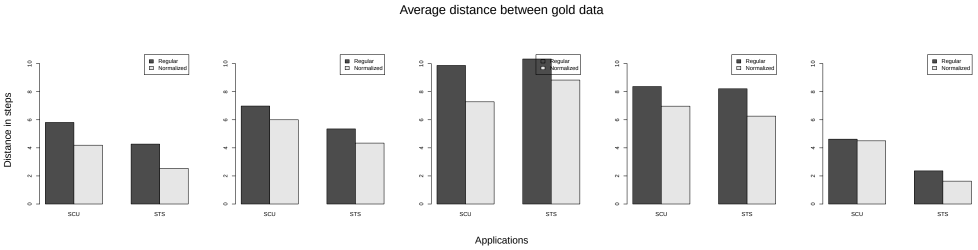


Figure 7.9: Average distance to gold annotation per app

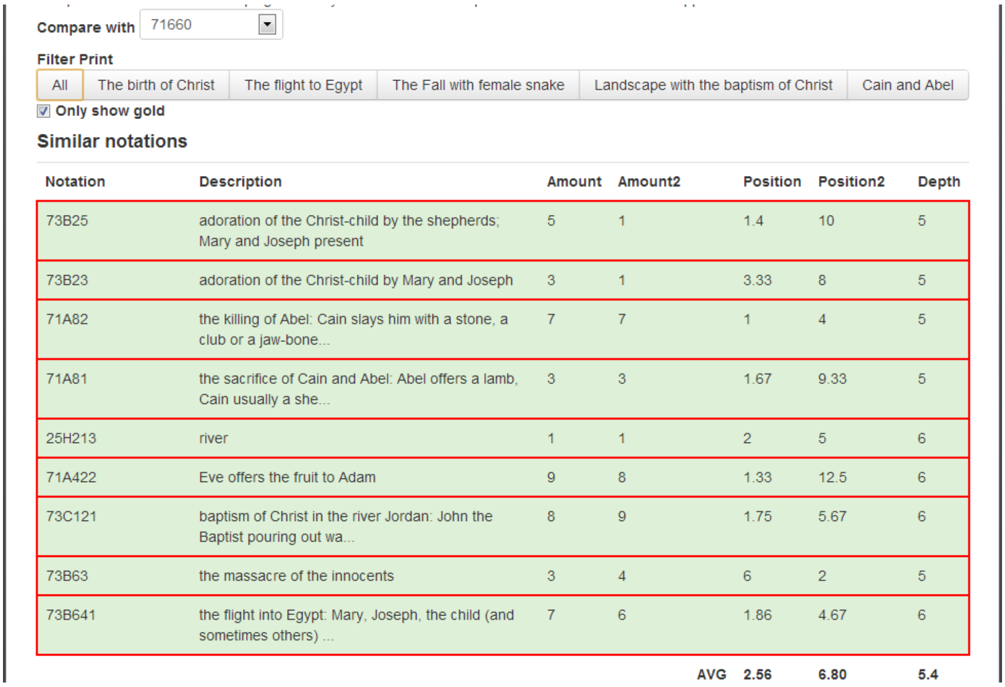
statistic it might favour the SCU.

If we look at the distance between user submitted and golden annotations in figure 7.9 it is clear that the results favour STS as its annotations are in closer proximity to the gold. SCU only scores better on print 3. In order to calculate these results we counted the steps on the taxonomy tree needed to get to the annotation and averaged them for each term over all possible gold annotations in order to get an average distance. We also, by lack of a better word, 'normalised' the results where we divided the distance by the number of times the term was added. As a result more popular terms will get a smaller distance penalty even though the distance might be large. However the results stayed largely the same. Both sorting strategies outperform the other on different aspects. There are reasons for both: terms in close proximity only differ in a small way and hence the results when searching with STS are grouped closer together. SCU favours terms by the RMA, thus gold annotations, and the results are a bit more scattered because the results have a larger variety. In order to make a better judgement we are going to take a closer look at the actual output of the sorting strategies in the next section.

7.4.2 Task Specific Statistics

In order to get a better understanding of the actual output of the plugins we used a plugin that details the position of an annotation when it was clicked. An example of the output of this plugin is shown in figure 7.10. Here we can see that SCU performs significantly better in giving a higher ranking to the correct gold terms, making users scroll less and offering a higher chance of being selected. The ranking for each term is an average number, because depending on the search keyword the ranking is not always the same. For the gold annotations that were similar fig. 7.10 shows that the gold annotations in SCU have an average ranking of 2,56 whilst STS has an average ranking of 6,80. As a result we can see a slight correlation between the times an annotation was added and the position of it. Of the six similar annotations that have a different amount of annotations, in five occasions the annotation that had a lower ranking was selected more often. For example five users of SCU selected 73B25, opposed to one from STS, likely because the term was hidden in the middle of the output in STS. Looking at all annotations (including non-gold) in figure 7.12 it can be seen that the ranking for SCU is better overall when compared to STS. However since this is a small dataset we refrain from drawing severe conclusions based on this. In our experiment we opted to display a maximum of 20 results at once to a user. The results would likely have been the same had we only shown 10, but limiting the choices to five would severely favoured SCU.

Another metric plugin gives an overview of the searches made by the users and the resulting annotations. As can be seen in figure 7.11 73B641 was added three times, but all from different searches and rankings. The term was added 13 times in total, but originated from 8 different searches. This gives a good idea which keywords work for a particular method and which don't. As we can see *egypt* works for both applications whereas *mary and joseph* and *flight* only seem to work for SCU with regards to that particular application. What this plugin doesn't show yet, but would be a helpful addition are the searches that didn't result in an annotation and if the term was visible but wasn't clicked.


Compare with 71660

Filter Print

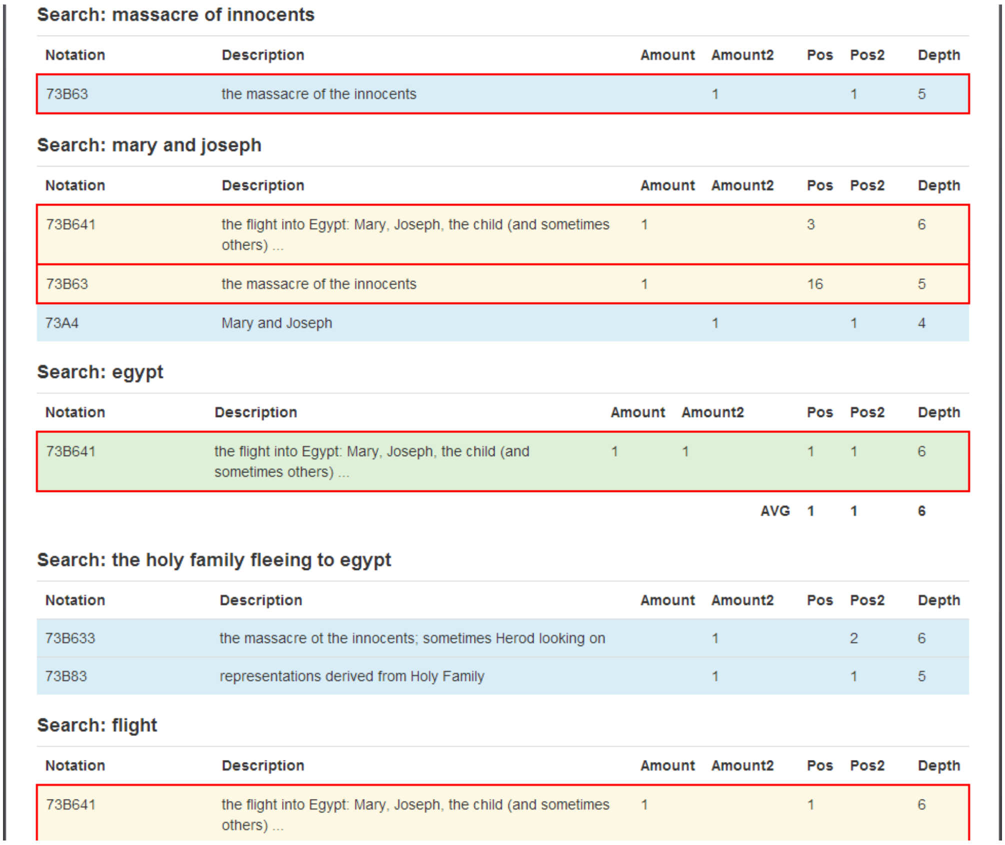
All
The birth of Christ
The flight to Egypt
The Fall with female snake
Landscape with the baptism of Christ
Cain and Abel

☑ Only show gold

Similar notations

Notation	Description	Amount	Amount2	Position	Position2	Depth
73B25	adoration of the Christ-child by the shepherds; Mary and Joseph present	5	1	1.4	10	5
73B23	adoration of the Christ-child by Mary and Joseph	3	1	3.33	8	5
71A82	the killing of Abel: Cain slays him with a stone, a club or a jaw-bone...	7	7	1	4	5
71A81	the sacrifice of Cain and Abel: Abel offers a lamb, Cain usually a she...	3	3	1.67	9.33	5
25H213	river	1	1	2	5	6
71A422	Eve offers the fruit to Adam	9	8	1.33	12.5	6
73C121	baptism of Christ in the river Jordan: John the Baptist pouring out wa...	8	9	1.75	5.67	6
73B63	the massacre of the innocents	3	4	6	2	5
73B641	the flight into Egypt: Mary, Joseph, the child (and sometimes others) ...	7	6	1.86	4.67	6
			AVG	2.56	6.80	5.4

Figure 7.10: Average depth in the search results list


Search: massacre of innocents

Notation	Description	Amount	Amount2	Pos	Pos2	Depth
73B63	the massacre of the innocents		1		1	5

Search: mary and joseph

Notation	Description	Amount	Amount2	Pos	Pos2	Depth
73B641	the flight into Egypt: Mary, Joseph, the child (and sometimes others) ...	1		3		6
73B63	the massacre of the innocents	1		16		5
73A4	Mary and Joseph		1		1	4

Search: egypt

Notation	Description	Amount	Amount2	Pos	Pos2	Depth
73B641	the flight into Egypt: Mary, Joseph, the child (and sometimes others) ...	1	1	1	1	6
			AVG	1	1	6

Search: the holy family fleeing to egypt

Notation	Description	Amount	Amount2	Pos	Pos2	Depth
73B633	the massacre of the innocents; sometimes Herod looking on		1		2	6
73B83	representations derived from Holy Family		1		1	5

Search: flight

Notation	Description	Amount	Amount2	Pos	Pos2	Depth
73B641	the flight into Egypt: Mary, Joseph, the child (and sometimes others) ...	1		1		6

Figure 7.11: Annotations resulting from a search

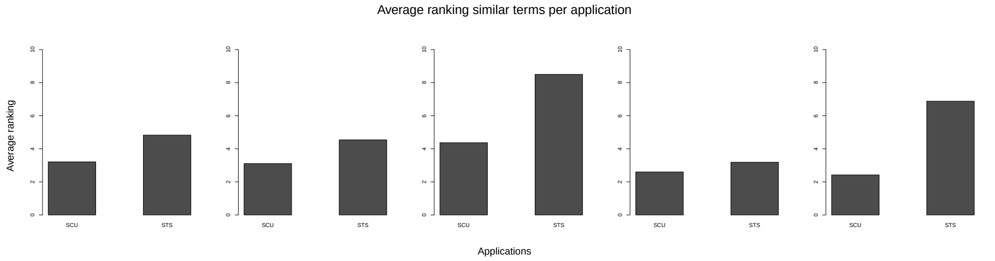


Figure 7.12: Average ranking similar terms per application. Lower is better

7.5 Remarks about the evaluation

In the previous sections we tried to compare SCU with STS. However the goal of this exercise wasn't to draw a conclusion which strategy was better, it was to see if such a conclusion could be made by a curator. And we believe this is indeed the case. Although it is hard to draw severe conclusions due to the small scale of the experiment, we believe that a curator would lean towards SCU as the metrics show it is slightly faster and provides more annotations that match the gold. However, because this method uses data used by the RMA it is only effective in an area of prints which already have been annotated by the RMA's professionals. If users need to annotate prints in an unannotated area the crowd annotators would suffer from a cold start problem and SCU would perform similar to STS. (This might explain the differences small differences in our experiment, when users search for terms not used by the RMA).

We have seen that the workbench is capable of successfully executing and comparing two strategies. Although we tried hard to make the workbench compatible with many other strategies we cannot guarantee that it is, as the workbench is created with the use case of section 5.3 in mind. Therefore we need to perform a more elaborate evaluation that evaluates the properties of the workbench. In figure 5.2 we made an initial list that aims to cover all different categories of possible strategies; however a more elaborate study needs to be performed. Based on this study a developer should make an implementation for each strategy category. If every category is implementable by the workbench we can conclude that the workbench has been created in a sound manner.

Furthermore a study needs to be performed using the curators as it needs to be determined that they can indeed configure their desired strategies using different datasets and plugins. Afterwards it is important to investigate if they are capable of interpreting the results in a similar way as we do. This can be done using a user study where their actions are monitored together with a questionnaire that asks for additional information. This could be feedback like how useful they found certain features or how happy they are with the tool.

When we look at the annotations obtained by our strategies with regards to quality, like we did in table 5.1, we can rate the coverage, the portion of the print that has been described, as high. Because there is some overlap between annotations, as some prints have been annotated with slightly similar annotations due to broader or narrower relations, Least-Effort has received a medium score. This is because Least-Effort states that the number of annotations for identifying an object should be minimised. In some cases a particular annotation was added by all lay annotators, so the popularity is high whilst Uniformity and Exclusion are added automatically by using a taxonomy.

Chapter 8

Conclusions and Future Work

This chapter concludes the work done in this thesis. In section 8.1 answer the research questions we posted in the beginning of this thesis and in section 8.2 we will summarise the made contributions. Finally we give some pointers for future work in section 8.3.

8.1 Conclusions

In this section we give a summary of the answers to the research questions we asked ourselves in the beginning of this thesis:

RQ1 What are the annotation processes of crowd annotators for multimedia data?

In order to answer this question we made an overview of different annotation processes and systems used by the crowd, using both systems which are used in a general and a CH setting. We found that most systems use a folksonomy which lets the crowd annotate using any term they want, although efforts are being made to maintain a uniform collection of terms by actively (auto-)suggesting terms.

RQ2 What are the annotation processes of Cultural Heritage professionals for digitised collections?

Based on research and interviews we documented the process off the RMA which should be representative for other CH institutes. In order to disclose the content of a print they use the Iconclass taxonomy to make sure their terms are of high quality. Their current problem is that they have constrained resources and as a result they are unable to annotate every detail they want.

RQ3 Which steps are needed to allow crowd annotators to annotate on a professional level?

The biggest difference we found is that the crowd annotates using systems based on a folksonomy, whereas at the RMA uses the Iconclass taxonomy to ensure their quality demands. Therefore it became clear that if we wanted to let the crowd annotate, a big step towards approximating the RMA's quality was using the same taxonomy.

However because a taxonomy might be hard for a non-professional we created a process consisting of five Annotation Steps which should help making the taxonomy more accessible. A combination of five steps creates a unique strategy.

RQ4 How can we support curators with rapid creation and evaluation of different strategies for these steps?

Because *the* perfect strategy doesn't exist and varies based on the requirements, we need a system which allows a CH curator to quickly prototype and test strategies. As a result we created the workbench which uses plugins to add functionalities. A combination of plugins, according to our five Annotation Steps, will represent a strategy. The goal was make the creation of an experiment as simple as possible, whilst still having powerful plugins. During the experiment metric plugins monitor and visualise the user's actions, allowing for an easy way to compare and evaluate different strategies.

During our evaluation of the workbench we have seen that lay contributors have no significant difficulties with annotating using a taxonomy whilst using our two strategies.

8.2 Contributions

The main contributions made in this thesis can be summarised as follows:

1. An overview of different types of annotations and their usages
2. Different processes for making annotations:
   - A generic crowd annotation process of how the crowd currently makes annotations
   - A modification of that process that allows it to be used in a taxonomy setting
   - A possible RMA process which includes the workbench as a testing platform
3. A workbench for comparing and gaining insight into different annotation strategies
   - The workbench has a working implementation and can be used in a CH setting
   - Extensible with the use of our custom plugin system
   - Easy to use for a curator
4. An initial evaluation of the Workbench and its capabilities

8.3 Future work

As we mentioned in section 7.5 further evaluation is necessary with regards to the workbench its ability to implement strategies and the ability of a curator to work with the workbench. Apart from the workbench, future work also is needed in other areas. For example during this thesis we hardly mentioned the semantic web, which makes it easier to reason about prints. By using a taxonomy for annotations we've made a big step towards the usage of the semantic web, as the taxonomy makes it easier to connect to existing ontologies. By using doing so we can reason about prints and deduct that prints annotated with "woman" also contain "humans".

One of the things we struggled with during this thesis was determining the correct metrics for comparing two different strategies to each other. Since a curator will likely have the same struggles the process of determining metrics should be supported. After the correct metrics have been determined, it is also important to visualise the results in a proper way, numbers mean nothing if they can't be understood by the curator.

During this thesis we mainly talked about images, or objects represented by one, but in our introduction we also mentioned that we took a couple of videos on our hypothetical trip to New York. These videos could also be annotated which allows us to find out what happens at specific time during the video. It should be possible to extend the workbench with so called temporal annotations, but it is likely that some of the processes should be revised.

Each annotation has equal weight when an annotation has been made, which might not be helpful for retrieval purposes. Currently it is possible to rank objects by the percentage of matched annotations. Say a print of a landscape with a farm has a fox in the background and as such it has five annotations and one of them is "fox". Another print is a full frame drawing of a fox and therefore only has the "fox" annotation. Seen in percentages this means that the first print is 20% fox and the other 100% and therefore should be ranked as such. But since people are already manually annotating each object, a useful addition could be to let them draw a rectangle around the fox, which should give a more accurate percentage that can be used for sorting the prints.

Bibliography

[1] L. Aroyo. Accurator: Ask the right crowd, enrich your collection. Presented at the SealincMedia symposium, 2013. http://www.slideshare.net/laroyo/sealincmedia-symposium-2013.

[2] Johan Oomen and Lora Aroyo. Crowdsourcing in the cultural heritage domain: opportunities and challenges. In *Proceedings of the 5th International Conference on Communities and Technologies*, pages 138–149. ACM, 2011.

[3] Jeff Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

[4] Jeff Howe. Crowdsourcing: A definition. *Crowdsourcing: Tracking the rise of the amateur*, 2006.

[5] Manish Gupta, Rui Li, Zhijun Yin, and Jiawei Han. Survey on social tagging techniques. *ACM SIGKDD Explorations Newsletter*, 12(1):58–72, 2010.

[6] Panagiotis Ipeirotis. Demographics of mechanical turk. *CeDER-10-01 working paper*, 2010.

[7] Alessandro Bozzon, Marco Brambilla, Stefano Ceri, and Andrea Mauri. Reactive crowdsourcing. In *Proceedings of the 22nd international conference on World Wide Web*, pages 153–164. International World Wide Web Conferences Steering Committee, 2013.

[8] Sabine Buchholz and Javier Latorre. Crowdsourcing preference tests, and how to detect cheating. *Proc. Interspeech2011, Florence*, 2011.

[9] Seth Earley. Folksonomy versus taxonomy, 2007. http://www.earley.com/blog/folksonomy-versus-taxonomy Retrieved on 10/04/13.

[10] TopQuadrant$^{\text{TM}}$. Controlled vocabularies, taxonomies, and thesauruses (and ontologies). http://www.topquadrant.com/docs/whitepapers/cvtaxthes.pdf Retrieved on 15/04/13.

[11] Thomas Vander Wal. Folksonomy definition and wikipedia, 2005. http://www.vanderwal.net/random/entrysel.php?blog=1750 Retrieved 15/04/13.

[12] Bob J Wielinga, Guus Schreiber, Jan Wielemaker, and JAC Sandberg. From thesaurus to ontology. In *Proceedings of the 1st international conference on Knowledge capture*, pages 194–201. ACM, 2001.

[13] Céline Van Damme, Martin Hepp, and Katharina Siorpaes. Folksontology: An integrated approach for turning folksonomies into ontologies. *Bridging the Gap between Semantic Web and Web*, 2:57–70, 2007.

[14] Christoph Trattner, Christian Körner, and Denis Helic. Enhancing the navigability of social tagging systems with tag taxonomies. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, page 18. ACM, 2011.

[15] Murtha Baca and Patricia Harpring. Categories for the description of works of art (cdwa): List of categories and definitions., 2011. http://www.getty.edu/research/publications/electronic_publications/cdwa/definitions.pdf Retrieved 05/05/13.

[16] Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 140, 1932.

[17] Zhichen Xu, Yun Fu, Jianchang Mao, and Difu Su. Towards the semantic web: Collaborative tag suggestions. In *Collaborative web tagging workshop at WWW2006, Edinburgh, Scotland*, 2006.

[18] Pere Obrador, Xavier Anguera, Rodrigo de Oliveira, and Nuria Oliver. The role of tags and image aesthetics in social image search. In *Proceedings of the first SIGMM workshop on Social media*, pages 65–72. ACM, 2009.

[19] Jennifer Trant, Bruce Wyman, et al. Investigating social tagging and folksonomy in art museums with steve. museum. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland*, 2006.

[20] A Greg. Your paintings: public access and public tagging. *Journal of the Scottish Society for Art History*, 16:48–52, 2011.

[21] Chris Dijkshoorn, Jasper Oosterman, Lora Aroyo, and Geert-Jan Houben. Personalization in crowd-driven annotation for cultural heritage collections. *4th International Workshop on Personalized Access to Cultural Heritage PATCH 2012, Montral, Canada, July 16-20, 2012*, 2012.

[22] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40. ACM, 2006.

[23] Michiel Hildebrand, Jacco Van Ossenbruggen, Lynda Hardman, and Geertje Jacobs. Supporting subject matter annotation using heterogeneous thesauri: A user study in web data reuse. *International Journal of Human-Computer Studies*, 67(10):887–902, 2009.

[24] Emanuele Quintarelli. Folksonomies: power to the people. 2005. http://www.iskoi.org/doc/folksonomies.htm Retrieved on 04/06/13.

[25] Ching-Chieh Kiu and Eric Tsui. Taxofolk: A hybrid taxonomy–folksonomy structure for knowledge classification and navigation. *Expert Systems with Applications*, 38(5):6049–6058, 2011.

[26] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117, 1998.

[27] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[28] Tom Tullis and Bill Albert. *Measuring the user experience: Collecting, Analyzing, and Presenting Usability Metrics*. Elsevier Inc., first edition, 2008.

[29] Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M Henne. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, 18(1):140–181, 2009.

[30] Brian Mulloy. *Web API Design: Crafting Interfaces that Developers Love*. Apigee, 2012. http://offers.apigee.com/web-api-design-ebook/.

Appendix A

Complete CDWA Analysis

#	Element Category	Suitable for Externals
1	Object/Work Core	No
2	Classification Core	No
3	Titles or Names Core	No
4	Creation Core	No
5	Styles/Periods/Groups/Movements	Maybe
6	Measurements Core	No
7	Materials and techniques Core	No
8	Inscriptions/Marks	No
9	State	No
10	Edition	No
11	Facture	No
12	Orientation/Arrangement	No
13	Physical Description	No
14	Condition/Examination history	No
15	Conservation/Treatment history	No
16	Subject matter Core	Yes
17	Context	No
18	Descriptive note	Maybe
19	Critical responses	No
20	Related works	No
21	Current Location Core	No
22	Copyright / Restrictions	No
23	Ownership / Collecting history	No
24	Exhibition / Loan history	No
25	Cataloguing history	No
26	Related Visual Documentation	No
27	Related Textual References Core	No
28	Person/Corporate Body Authority Core	No
29	Place / Location Authority Core	No
30	Generic Concept Authority Core	No
31	Subject Authority Core	No

Table A.1: List of top level categories of CDWA. For full explanations of these descriptions please see [15].

Appendix B

Plugin Sequence Diagram

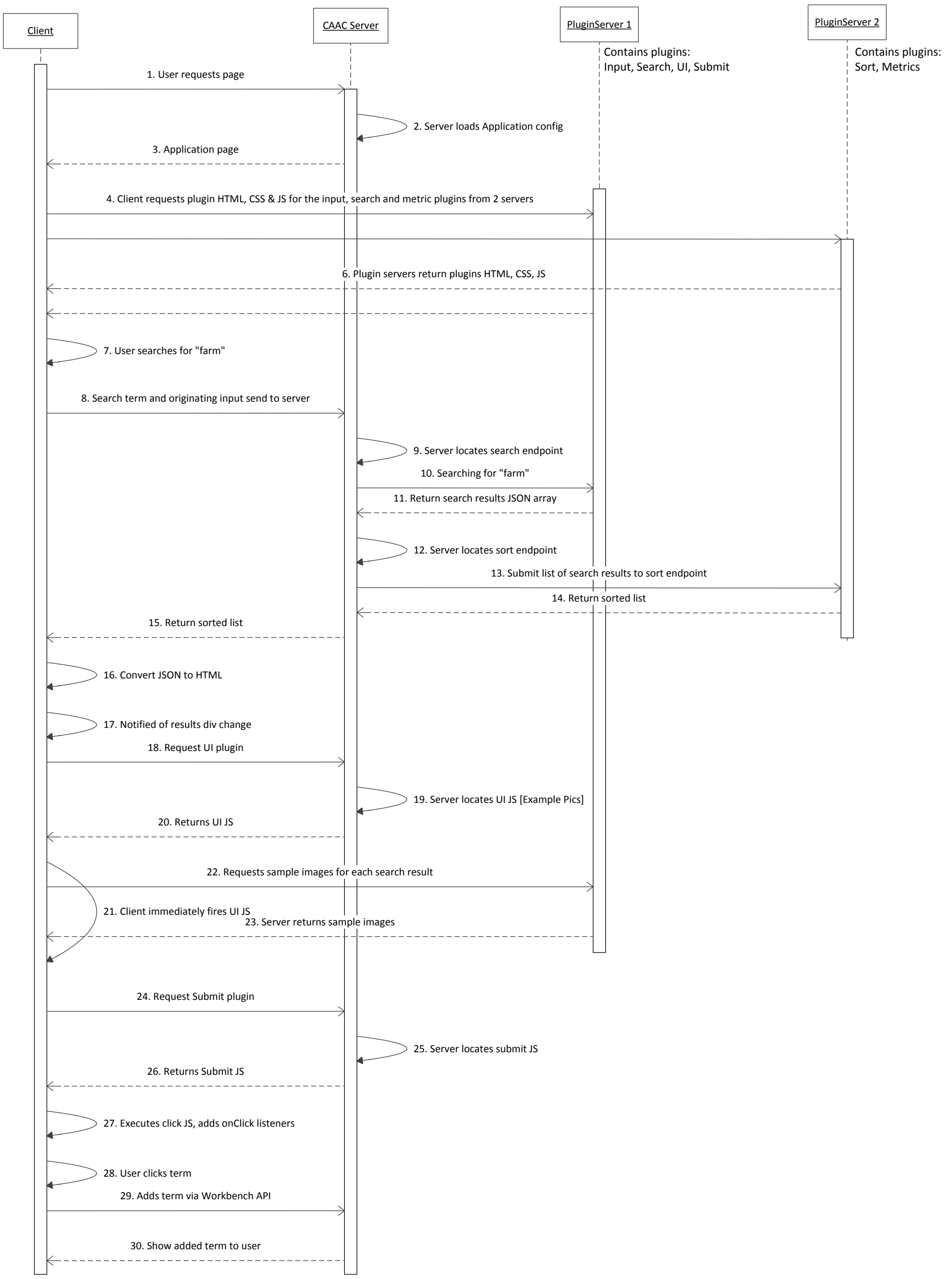

Client
CAAC Server
PluginServer 1
PluginServer 2
Contains plugins:
Input, Search, UI, Submit
Contains plugins:
Sort, Metrics
1. User requests page
2. Server loads Application config
3. Application page
4. Client requests plugin HTML, CSS & JS for the input, search and metric plugins from 2 servers
6. Plugin servers return plugins HTML, CSS, JS
7. User searches for "farm"
8. Search term and originating input send to server
9. Server locates search endpoint
10. Searching for "farm"
11. Return search results JSON array
12. Server locates sort endpoint
13. Submit list of search results to sort endpoint
14. Return sorted list
15. Return sorted list
16. Convert JSON to HTML
17. Notified of results div change
18. Request UI plugin
19. Server locates UI JS [Example Pics]
20. Returns UI JS
22. Requests sample images for each search result
21. Client immediately fires UI JS
23. Server returns sample images
24. Request Submit plugin
25. Server locates submit JS
26. Returns Submit JS
27. Executes click JS, adds onClick listeners
28. User clicks term
29. Adds term via Workbench API
30. Show added term to user

Figure B.1: Sequence diagram of interactions between plugins and two repositories

Appendix C

Plugin XML Structure

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="plugin">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="title" type="xs:string"/>
      <xs:element name="author" type="xs:string"/>
      <xs:element name="type">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="input"/>
            <xs:enumeration value="search"/>
            <xs:enumeration value="sort"/>
            <xs:enumeration value="ui"/>
            <xs:enumeration value="submit"/>
            <xs:enumeration value="data"/>
            <xs:enumeration value="metric"/>
            <xs:enumeration value="general"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="version" type="xs:float"/>
      <xs:element name="json_endpoint" type="xs:anyURI"/>
      <xs:element name="js" type="xs:string"/>
      <xs:element name="js_admin" type="xs:string"/>
      <xs:element name="css" type="xs:string"/>
      <xs:element name="css_admin" type="xs:string"/>
      <xs:element name="html" type="xs:string"/>
      <xs:element name="html_admin" type="xs:string"/>
      <xs:element name="description" type="xs:string"/>
      <xs:element name="dependencies">
```

```
        <xs:complexType>
          <xs:sequence>
            <xs:element name="dependency" type="xs:string"
                minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="parameter" minOccurs="0"
                maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="title" type="xs:string"/>
                  <xs:element name="description"
                      type="xs:string"/>
                  <xs:element name="placeholder"
                      type="xs:string" minOccurs="0"/>
                  <xs:element name="type" minOccurs="0">
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:enumeration value="password"/>
                        <xs:enumeration value="radio"/>
                        <xs:enumeration value="checkbox"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element name="key" type="xs:string"/>
                  <xs:element name="values">
                    <xs:complexType mixed="true">
                      <xs:sequence>
                        <xs:element type="xs:string" name="value"
                            minOccurs="0" maxOccurs="unbounded"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
</xs:schema>
```

*Restrictions values, lengths etc. are omitted for legibility when not deemed vital for the understanding of the reader.