Technical University of Denmark

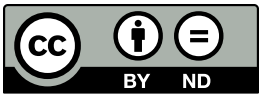
Master's thesis

Crowdsourcing Platform for Museums

Kræn Vesterberg Hansen
M.sc.eng. Computer Science and Engineering
s082932@student.dtu.dk
Spring 2014

DTU Compute
Department of Applied Mathematics and Computer Science

Kræn Hansen: Crowdsourcing Platform for Museums — Discovering, designing and integrating a service oriented architecture to support a museum, when getting help from individuals within its community to produce digital machine-readable content about its collections.



Abstract

This thesis addresses a strategic challenge at National Museum of Denmark to engage with external people, interested in contributing information about their collection of more than half a million coins and medals. This approach of getting outsiders to help with the completion of many small tasks are popularly known as crowdsourcing. This entails a need for the transcription of handwritten protocols, establishment of references between of entries in protocols and photographs of coins. These coins also references both structured and non-structured metadata.

Does a digital platform for crowd engagement, in the museum's context, exist? And how is such a platform integrated with the existing infrastructure of the museum? The report considers the MediaWiki, Amazon's Mechanical Turk and Zooniverse's Scribe transcription interface, and finds that the MediaWiki fits approximately 70% of the requirements.

Existing cases of successful crowdsourcing projects, national as well international is mentioned and the solution builds upon APIs of existing infrastructure components (such as the existing collection management system GenReg Mønt and the Canto Cumulus digital asset management system) in a modular and reusable architecture.

The report approaches the challenge in a three part process, greatly inspired by the software process model of "Reuse-oriented software engineering" proposed by Professor of Software engineering at the University of St Andrews, Ian Summerville.

Part I: It establishes a preliminary product backlog of user stories, defining the requirements specification.

Part II: The system fitness analysis, estimates the fitness of the before mentioned systems to categories of features from the requirements specification, through a conversion of estimated workload, established through the planning poker game.

Part III: Presents the design and implementation of nine user stories, ultimately presenting a graphical user interface for transcription of protocols in a MediaWiki as well as an integration between the MediaWiki and the Canto Cumulus DAMS.

It is concluded that the designed and implemented solution supports the essential activities required of a platform for crowd engagement in the context of the National Museum of Denmark.

Resumé

Denne afhandling omhandler en strategisk udfordring ved Nationalmuseet, der ønsker at engagere eksterne folk, der er interesserede i at bidrage med oplysninger om deres samling af mere end en halv million mønter og medaljer . Fremgangsmåde med at for eksterne til at gennemføre mange små opgaver er populært kendt som crowdsourcing. Dette medfører et behov for transskribering af håndskrevne protokoller, etablering af referencer mellem protokoller og fotografier af mønter. Disse mønter refererer også både strukturerede og ikke- strukturerede metadata.

Eksisterer der en digital platform til engagement af frivillige, der passer til museets situation? Og hvordan kan en sådan platform integreres med den eksisterende infrastruktur ved museet? Rapporten behandler MediaWiki, Amazons Mechanical Turk og Zooniverse's Scribe transskriptions brugergrænseflade, og finder, at MediaWiki opfylder op imod 70% af kravene.

Eksisterende tilfælde af vellykkede crowdsourcing-projekter, nationalt såvel som internationalt er nævnt, og løsningen bygger på API'er udstillet af komponenterne i den eksisterende infrastruktur (såsom det eksisterende genstands-registreringssystem GenReg Mønt og Canto Cumulus - et system til håndtering af digitale photografier) i en modulær og genbrugelig arkitektur.

Rapporten behandler udfordringen i en tre delt proces, der i høj grad er inspireret af en software proces model kaldet "Genbrugs-orienteret software engineering" der er foreslået af professor i Software engineering ved University of St Andrews, Ian Summerville.

Del I: Etableringen af en foreløbig product backlog, indeholdende brugsmønstre (user stories), dette udgør kravspecifikationen.

Del II: En analyse anslår egnethed af de før nævnte systemer til at opfylde kategorier af funktioner fra kravspecifikation, gennem en konvertering af skønnede arbejdsbyrde, der etableres gennem planlægnings spillet planning poker.

Del III: Design og implementering af funktionalitet der løser ni brugsmønstre præsenteres, i sidste ende præsenteres en grafisk brugergrænseflade til transskription af protokoller i et MediaWiki samt en integration mellem MediaWiki og Canto Cumulus DAMS.

Det konkluderes, at den designede og implementerede løsning understøtter de væsentlige aktiviteter, der kræves af en platform for involvering af frivillige i digitaliserings processen ved Nationalmuseet.

Preface

This master's thesis was prepared in the fulfilment of the requirements for acquiring a Master's degree of Science in Engineering / Computer Science and Engineering. The equivalent of the Danish title "Cand.polyt. i Informationsteknologi".

The work was supervised by associate professor Hubert Baumeister at the Technical University of Denmark and assigned a workload of 35 ECTS credits.

This thesis deals with elicitation of requirements for a software system, analysis of fitness of existing technical solutions as well as design and implementation of a software system. It is to be used when a museum generates metadata about its artefacts, from the engagement with individuals within its community. This process of utilizing a large community to perform many small tasks is generally know as crowdsourcing.

The thesis is based on an actual need for a software system at the National Museum of Denmark, where a collection of approximately half a million coins and medals are photographed and protocols explaining their origin scanned, for manual transcription. My primary point of contact and product owner at the museum was Jacob Riddersholm Wang.

Acknowledgements

I want to thank the National Museum of Denmark, for trusting me with a project of such a large strategic importance. A warm-hearted thanks goes out to the employees (Charlotte S.H. Jensen, Helle Horsnaes and Bodil Qvistgaard) at the museum, and especially Jacob Riddersholm Wang for demonstrating a highly professional, yet personal approach to my project, and for always responding rapidly and in great detail whenever I had a question or requested anything related to the project.

I want to thank my main supervisor Hubert Baumeister, that despite our disagreements has demonstrated patience and academic insights, although I have not been utilizing his guidance as much as I originally expected.

A special thanks to the crowd of the four numismatics Niels Jørgen Jensen, Mogens Skjoldager, Lars Christensen and Preben Nielsen from the Danish Numismatic Association, and the association as such for maintaining an interest in the narrow subject of coins an medals, around which the association's members keep our country's history alive by telling the stories behind coins for our future generations to learn from past mistakes.

A very special thanks to my girlfriend and academic sparring partner Camilla Christensen – your ears and shoulders have been irreplaceable when writing the thesis as an individual. A warm thanks to my family and friends who have been subject to my lack of communication during busy periods of the project. Especially my sister Gunilla Vesterberg for help correcting formulations and spell checking the report.

A final thanks to my business partner Christian Høeg and all of my co-working employees at BIT BLUEPRINT ApS for showing great interest and indulgence throughout the writing of my thesis: Malthe Jørgensen, Sandra Rose Cliff, Joachim Jensen, Mads Lundt, Jens Christian Hillerup, Markus Færevaag and Attila Sukosd.

Contents

Introduction

The project was first introduced to me as a blog post on the online weblog of the department of digitization at the National Museum of Denmark. The post was titled "Digitalisering og crowdsourcing af møntsamlingen" [29] and it described how the museum would like to engage with external people, interested in generating data about their collections of more than half a million coins and medals.

Public galleries, libraries, archives and museums have a hard time finding the resources and knowledge to generate content around their artifacts. This is why they would like to engage with people outside their institution, getting enthusiasts in their community to contribute content about their artifacts.

One such institution and one particular collection is the National Museum of Denmark's collection of coins. The museum expresses a need for the transcription of handwritten protocols (stories on the acquisition of particular coins). They express a need for an establishment of references between of entries in these protocols and high resolution photographs of coins. These coins are also supposed to have content around them, in both structured (particular features, weight, value, currency, etc) and non-structured (inscriptions, stories, etc) form.

An overall goal of the museum: Saving money on staff by engaging non-professional volunteers, from the outside of the institution, in the digitization process. A derived goal is getting contributions to their stack of technology to deliver more and interesting concepts to the general public. It is expected that crowdsourcing can be a driving force for technological development in general, as this is imposing new demands on the museum's IT infrastructure.

> The project answers the question: *Does a digital platform for crowd engagement, in the museum's context, exist? If yes: How is such a platform integrated with the existing infrastructure of the museum? If no: What is wrong with existing platforms? Suggesting a design for and partially implementing a solution.*

Examples of existing platforms, which could be considered for integration is the MediaWiki, Amazon's Mechanical Turk and other such platforms. Which platforms to use, depends on the type of content and the quality requirements that the museum has to the crowdsourced content.

As a supplement, I would like my research to consider existing cases of successful crowdsourcing projects, national as well as possibly international. This in order to derive best principles and commonly applicable characteristics, leading to their success.

The overall goal of my masters thesis will focus on solving the recurring problem of engaging enthusiasts (i.e. the crowd) in the process of generating and processing content about artifacts (i.e. metadata), in a sustainable and non-intrusive way. A solution should build upon APIs of existing infrastructure components (such as existing CM and DAM systems) in a modular and reusable architecture.

The benefits of a reuse-oriented approach to software engineering

I have approached the research question as a three-step process, greatly inspired by the abstract software process model of "Reuse-oriented software engineering" [38, page 35]. I will apply techniques and terms from agile software development in this process but the overall project model has a waterfall-like initial upfront investment of time, clarifying initial requirements and analysis of existing components.

When a software engineer approaches a challenge he has to keep in mind that the problem might have been solved before. In order for the engineer to propose an optimal result, some part of the project should

be invested in the exploration of existing components that might be solving the challenge elsewhere. This is increasingly the case with the rising number of open source project, containing reusable source code for components which may be modified or deployed directly, without paying any license fees.

The approach to software engineering practice which is typically taught at the universities almost always has a white-canvas design brief. No code has been written - design and implement a system from ground up. But this is very rarely the actual case as an organization has existing infrastructure consisting of web services and other applications - and the best solution at the least amount of resources (maximizing the return of business value on investment of engineering hours) might very well be full or partial reuse of existing components. This is an issue - as the computer scientists leaving education has limited methods and techniques to help them investigate and learn the architecture of an existing code-base. When the university fails to educate software engineers, with a reuse-oriented approach, the university is essentially producing self-proclaimed geniuses which initial response to any project is, let's build it from scratch - because they have no tools to evaluate the risk of building software on-top of or by modification of a particular exisiting system.

Software has the remarkable property that it can be reproduced at a very low cost. The source code of an arbitrarily large software system can be copied within seconds and if it has been designed for reuse, guides exists that can enable an installation or deployment of the software within minutes.

The reuse-oriented software engineering differentiates itself from the classical white-canvas approach, by the need for this upfront investment in the investigation of components which are fair to assume as potential candidates for reuse in the particular context of the challenge. Besides this, the process can be a regular agile software development process in sprints iteratively delivering increments in the product. This is essentially a trade-off, investigating existing components in the hopes of saving time on design and implementation.

Ian Summerville has the following perspective on the benefits of reuse-oriented software engeneering:

"An obvious advantage of software reuse is that overall development costs should be reduced. Fewer software components need to be specified, designed, implemented, and validated. However, cost reduction is only one advantage of reuse. In Figure 16.1, I have listed other advantages of reusing software assets. However, there are costs and problems associated with reuse (Figure 16.2). There is a significant cost associated with understanding whether or not a component is suitable for reuse in a particular situation, and in testing that component to ensure its dependability. These additional costs mean that the reductions in overall development costs through reuse may be less than anticipated." [38, page 427]

Approaching the question

The report is split into three parts, each representing the different phases of the project.

Part I — Requirements

I want to answer if *a digital platform for crowd engagement, in the museums context, exists.* For me to do this, part of the challenge is understanding *the museums context.*

This part of my thesis will provide an understanding the requirements for a digital platform for crowd engagement in the museums context. Proposing functional and non-functional (also called qualitative) requirements for the solution to-be.

Sommerville calls this step “Requirements specification” [38, page 35], “Software specification”[38, page 36], “Requirements analysis and definition” [38, page 31] and “The requirements engineering process” [38, page 38] interchangeably.

I have chosen that the output of this part is an initial product backlog, as known from the agile SCRUM methodology [23, page 5]. The backlog consists of product backlog items, which I have chosen to primarily formulate as user stories [47], prioritized in the order of perceived business value, from the museums product owner's point of view.

Part II — System fitness analysis

Once the preliminary requirements for the system is known, it should be possible to answer the first part of my research question: *Does a digital platform for crowd engagement, in the museums context, exist?*

I will select a set of existing systems and components and analyze each of them to estimate their fitness towards the elicited requirements. Proposing a candidate system for integration/expansion as well as an analysis of the requirements missing fulfillment or changed if the proposed system is chosen as foundation when designing and implementing a solution.

Sommerville calls these step “Component Analysis” [38, page 35] and “Requirements modification” [38, page 35] respectively.

Because the preliminary backlog is potentially a long list of requirements, and in order for me to raise the level of abstraction, I will categorize the functional requirements in categories of ideally orthogonal subjects, before estimating the fitness of the candidate systems.

To reduce bias and to increase precision on the estimates, I will be estimating workload if the feature were to be implemented, instead of fitness. This is because processes such as the planning poker game, already exists and are a trusted method for estimation of workload in practicing SCRUM teams, on a daily basis.

Part III — Designing and implementing a solution

I will then be engaging in a design and implementation phase, where the system chosen in the fitness analysis will serve as the basis on-top of which the modified requirements will be implemented.

I will be designing and implementing only one sprint, without the feedback of users, as the result of the thesis is not a full fledged software system, as much as a first working prototype, which could be used when proving the business opportunity.

Sommerville proposes a separation in stages between design and development when performing reuse-oriented software engineering [38, page 35] but I have chosen a variant of an agile method for this. Focusing on each of the user stories of the modified requirements one at a time, in the sequence of perceived business value from the museums product owners point of view.

My definition of done, i.e. when I continue to the next user story, is that it has to be documented in the report, but automated tests has not been the focus of my project.

Part I

Requirements

"The formulation of the problem is often more essential than its solution, which may be merely a matter of mathematical or experimental skill."

— Albert Einstein

1 Introduction to requirements elicitation

Why would You spend an upfront cost on eliciting requirements? After all – building something might give users an opportunity to tell you what they *don't* want.

When reusing software components to build a software system, the cost of choosing the wrong platform and changing to another might end up rendering everything designed and implemented up until that point of change, useless.

This is why the initial upfront cost of eliciting and analysing requirements will probably turn out to be a great investment.

With the challenge introduced and the approach defined in the previous section, this part of the process focuses on producing two main artefacts:

- **The existing software platform**
  An overview of the existing infrastructure and application systems already at the museum.
- **The product backlog**
  An ordered list of preliminary requirements for the system.

When finding out what is required of a software system it often pays of to prioritize requirements – after all not everything is equally important, so it would probably not be the best idea to just start in the corner first defined.

These requirements should not be seen as final. It is in itself a challenge to evaluate how much of the system requirements one has defined. In addition to this, the requirements of the system is a moving target. As a challenge gets redefined by stakeholders, priorities change within the organization and technology advances makes the user's expectations towards a software system change.

2 The goals of the museum

All too often it happens that an engineer is briefed with a partial suggestion of a solution to something the customer (or direct user) perceives as a problem. The problem being that the solution suggested might be very limited by the imagination of user and often very subjective, as the person formulating the brief is often also an actor within the solution to be implemented.

Some times understanding the motives behind the partially suggested solution can lead to other ways of solving the same challenge. Some of these incorporating requirements from other actors of the system better or reducing the cost of implementing the solution in an arbitrarily costly way. I.e. it often helps taking a step back and abstracting the concrete brief (often formulated as: "Build this system.") to an understanding of the goals and motives of the stakeholders.

One such stakeholder is the museum, and therefore its decision makers. They are measured on their ability to implement the strategic goals of the museum – proposing constrains on the project, which is not clearly visible when simply approaching a solution-oriented brief.

2.1 Strategic goals

How does a digital museum look in the information age? And how does the museum utilize the full potential of its increasingly digitized surroundings[1]?

The National Museum of Denmark has proposed an overall answer to this in their mission statement and vision[33].

Mission The National Museum holds and develops the conditions for everyone to gain insight into cultural history.

Vision The National Museum is recognized:

- for the ability to translate knowledge about the cultural history into experiences for all.
- as one of the leading museums in digital dissemination.
- for its role as a main museum.

It is therefore safe to assume that a strategic priority from the National Museum, is to become recognized as one of the leading museums in using digital tools processes internally as well as externally.

Another significant visionary goal is to be recognized for its role as a main museum, which the museum articulates[2] as a responsibility to share knowledge, data and technology with other museums around it.

2.1.1 Strategic guidance from the Ministry of Culture

As the museum is a recognized museum by the ministry of culture, the museum also has to follow the strategic guidance of the ministry, which includes the ministry's strategy on digitization[31].

"The investments, made in new IT systems, must lead to qualitative improvements and/or more effective business processes."[31, page 6, paragraph 1]

"The concept of culture in the digital context is about using digital media's strengths rather than reproducing what other media types are capable of. People need to be involved as users and also as producers of culture. Interaction with and among citizens on the web is central."[31, page 7, paragraph 3]

"Involving citizens in the development of new services is also an essential element in ensuring relevant and contemporary digital service. This applies both in the concept and development phase, but also when content to be produced. The opportunities to involve citizens when content is created, and data is made available to reuse in other contexts should always be investigated." [31, page 11, paragraph 2]

"Several of the cultural institutions are facing similar or identical challenges in the field. There is a need in the community to meet and address the challenges associated with acting as a governmental cultural institution. Challenges that often have a distinctive character in relation to the classical governmental institution. It is therefore essential that the resources and initiatives to the policy area, used to fully provide value to as many as possible."[31, page 17, paragraph 2]

[1] Such as smartphones, other digital museums and the internet in general.
[2] I have personally heard this through it's head of digitization, Jacob Riddersholm Wang.

2.1.2 The museums strategy on digitization

Jacob Riddersholm Wang, head of Digital Media at the National Museum of Denmark, identifies three general themes in the digitization strategy of the museum (2012-2015)[46]. The latter two addresses an optimization on the use of digital processes internally, as well as an increase in the general knowledge and terminology across all employees at the museum.

The first theme of the digitization strategy is "*Openness, availability and transparency*", stating that "*all objects, artefacts and photographs should be digitally available internally and externally in 2020. Everyone internal and external to the museum is enabled, through digital utilities and media, to add knowledge, thereby adding value to the collection, research and dissemination at the museum.*"[46, page 3]

The museum describes this strategic theme in 8 sub-goals:

1. *Making it easier to use the cultural history*: By licensing all photographs of documents and artefacts under a creative commons variant.
2. *To engage with the users where they are present*: Users might want to engage with the museums artefacts (their photos and metadata) via a Google search, via WikiPedia and other platforms not directly provided by the museum. Hand-held devices might as also be a preferred platform for users, when they are for example present at the museum.
3. *To include the user actively*: External users can contribute valuable information and help validate and possibly correct information registered by the employees at the museum. In the year 2014, a contained pilot project is conducted, where the users of the museum gets the possibility to contribute additional information and comments to objects and photos in the museums collections. Based on the experience, it is decided if such a solution can be permanently supported.
4. *Consolidated registration system for locations and conservation*: A consolidation of internal systems to prepare for a migration into a national system.
5. *Migration of the National museums databases to the "Fælles museums-it" system*: The latest news from the cultural ministry is that this system will be ready by 2016.
6. *To create a fully digital object database*: Missing data is acquired as appropriate data is the basis of the value proposition of an open access to data. Taking into account potential problems with sensitive and personal information.
7. *Digitization of the museums archivals and images*: The museum has large archives with registration cards and images which has partly been digitized. All registration cards and images are digitized into the digital assets management system Cumulus.
8. *Expansion of the research registration system PURE*: It is a goal to have the museums research results online via the PURE-portal, which is already deployed.

2.1.3 The origin of the project

It is safe to assume that the project is a direct consequence of the strategic third sub-goal "*to include the user actively*" in the strategic theme "*Openness, availability and transparency*".

Community manager at the museum Charlotte S. H. Jensen formulates an intent of creating a framed approach to crowdsourcing at the museum, in the initial blog post[29] presenting the intention of digitizing the collection.

"*It is the minority of Danish museums that has experience with crowdsourcing on a large scale, thus the project is a start on working more collaboratively with metadata.*" [29, paragraph 2].

Another basis for initiating the project is that the amount of coins is approximately between half a million and one million[29], which makes it improbable for the museum to register and categorize them all by themselves.

2.2 Stakeholders

This project, as any project, has stakeholders. The success of a system, is ultimately its ability to fulfil its vision, while respecting or taking advantage of the constraints posed by the context in which it will operate.

This is why a critical step in the process of successfully designing a software system, is to understand the pains, goals and constraints suggested by everyone involved and interested in the project. Basically understand the ecosystem in which the IT system will unfold.

The requirements presented and many decisions made throughout this thesis is based on qualitative interviews [3] with stakeholders, internal as well as internal to the museum:

- Head of digital media at the museum and product owner in this software project - Jacob Riddersholm Wang
- Curator at the museum - Helle Horsnaes
- Database developer at the museum - Bodil Qvistgaard
- Community manager at the museum - Charlotte S. H. Jensen
- Members of the crowd of numismatics, i.e. people that collect and research coins - Niels Jørgen Jensen, Mogens Skjoldager, Lars Christensen and Preben Nielsen

3 Related projects

Although crowdsourcing as a term is still increasingly gaining the attention worldwide[28], this is not the first time a museum or archive has turned to its community for a helping hand, with the digitization of it's collections of artefacts.

I have investigated a couple of these projects to see if any best practice or principles could be derived, which could guide the ideation phase when formulating requirements for this project. This also served as means of to the formulation of the concrete systems to use for my system fitness analysis, in part 2 of this report.

[3] The interviews has *not* been transcribed and is therefore not provided as appendices - as I have prioritized other sections of the thesis.

3.1 Politiets Registerblade (The police's register sheets)

According to the community, this is one of the first successful large-scale crowdsourcing projects in Denmark. It was stablished by Copenhagen City Archives in 2008, when the archive is contacted by the three largest national organisations within genealogy. The members of the crowd (of genealogists) need data about the relations between people for them to complete their respective family trees. This unique mutual benefit is a central driver of the motivation for the actors in the project, according to Jeppe Christensen [2, slide 8] – lead on the project, as well as the ability for the members of crowd to be able to compete on entering data.

The project reports a 60.000 volunteer hours to 6.000 development hours ratio, of which the archive only has expenses relating to development. [2, slide 6]

From a technical point of view, the project is based around a website, written en PHP on-top of the open source content management system Joomla. It is mentioned that their next project has to be generic [2, slide 15], as this project was build with the specific use case in mind. Below is a diagram showing the entities and relations in the system [2, slide 5].

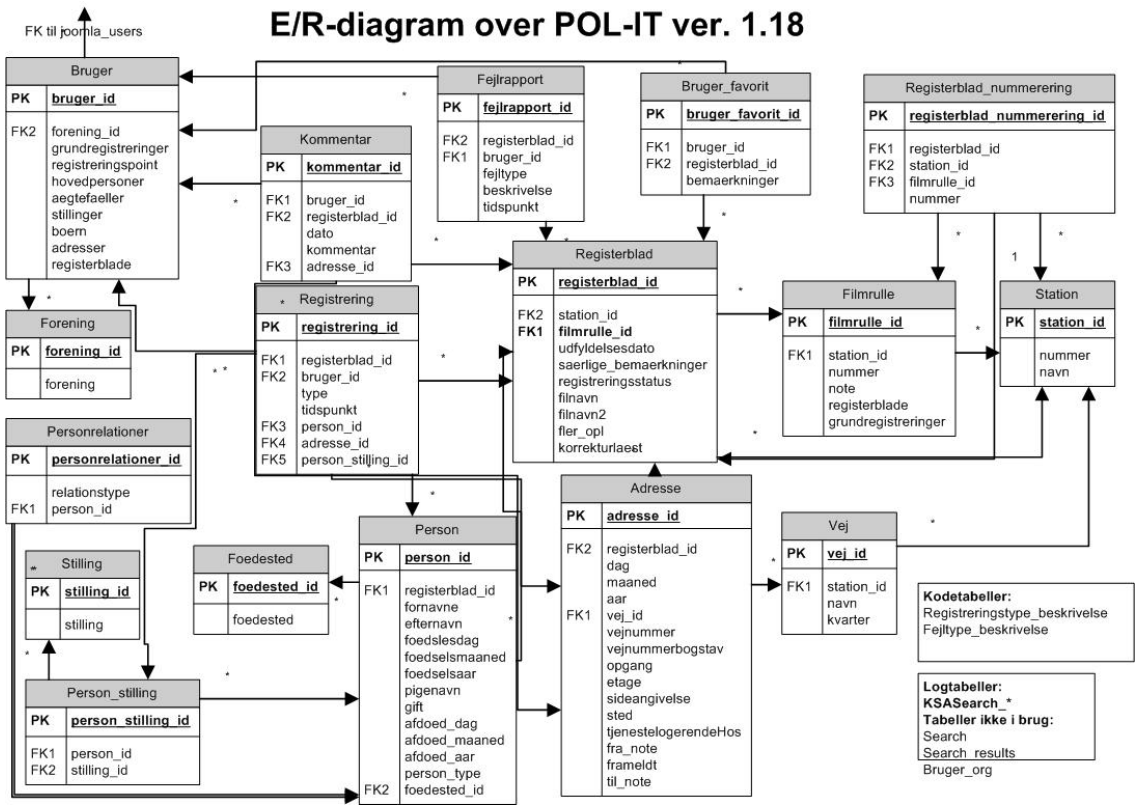


Figure 1: A diagram showing entities and relations within the "Politiets Registerblade" system. This diagram is not my work but courtesy of Jeppe Christensen.

As seen from the diagram - and from their statement of the need for a more generic product, this is probably not a suitable candidate for the fitness analysis (in part 2).

3.2 Danmark set fra Luften (Denmark seen from the air – before Google)

This crowdsourcing project is initiated by the The Royal Library, tagging and geographically locating aerial photos captured above Denmark from the year 1923 and onwards.

“The Royal Library’s collection of aerial photos consists of more than 3.5 million unique captures from 1923 up to 2009.” [34, following the "english" link].

The project was launched in September 2011 and in just two years the volunteers managed to pin more than 80% of the photographs to their physical location.

I am assuming that one of the reasons behind it’s success was its ability to attract people with local knowledge. And I have had conversations with people responsible at the library, mentioning its ability to provide incentive to the crowd members to look outside of their local knowledge, through the use of high-scores.

An interesting feature about the high-scores of the product is the ability to filter on geographical subsections of the country. So instead of looking who is the best in the whole country, the crowd members were able to compete within their geographical expertise.

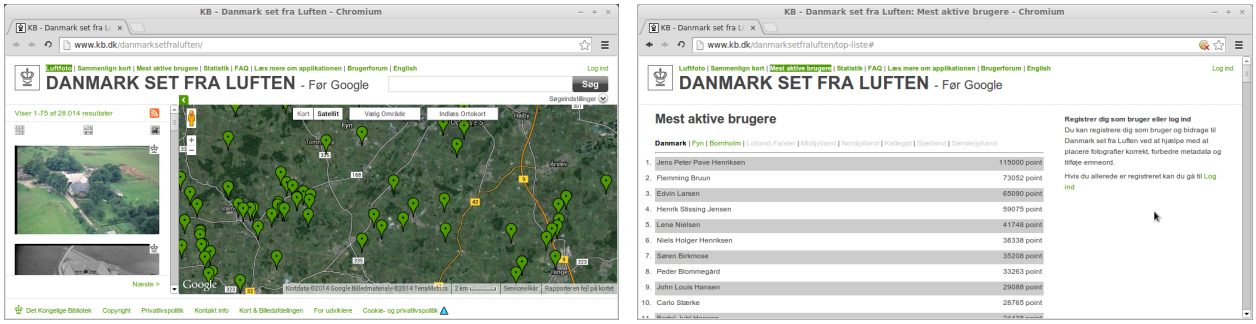


Figure 2: Screenshots of the “Danmark set fra luften”-product, with photos pinned to a map and the area-filtering high score.

The tagging of photos on a map is the central method of contribution on the library’s platform, so the central theme of the product does not directly apply to the museum’s project - thus is not an applicable subject for the analysis for re-usability.

3.3 Art Collector

Another more recent project, from spring 2013, is a study on “Crowdsourcing cultural heritage metadata through social media gaming” from the Department of Computer Science, Malmö University. [36]

The study introduces a range of different activities that the members of crowd can engage in as types of crowdsourcing in cultural heritage. Theoretical knowledge on the motivation of crowd engagement through the use of gamification is theme throughout the study. As the players of the implemented prototype game are expected to find an extrinsic motivation from the competition for non-monetary tokens and intrinsic motivation when receiving copies of pieces of artwork for their virtual collections.

Paraschakis calls this type of cultural heritage games, “Games with a purpose” (GWAP), a concept introduced by von Ahn & Dabbish [44].

This software has not been considered for reuse, mainly because it only provides tags of images, it is a prototype and not a finalized software product and finally because I couldn't find a link for the source code, thus i suspect that it has not been released for reuse under an open source license.

Figure 3: Screenshots from the Art Collector.

4 The existing software platform

It happens that projects a declared a success, although they end up leaving an organisation with broken work flows and frustrated employees. So trusting an organisation to determine if a software project is a success, might be politically biased. From a Darwinian point on software engineering, it is the survival of the fittest and not the most powerful individuals that survives. The competitive advantage of a software component, might very well be it's ability to fit within it's environment of existing components.

Some practice development of robust software systems through redundancy [27, 6, p. 7 / 81]. Although this might be useful for developers, I argue that this leads to confusion if introduced to end-users. If a software component provides functionality which is already provided by an existing component, people get confused on which to use and mistakes due to this ambiguity is inevitable. It might be a violation of the best practice to minimise redundant representations of data in a data model, which could have been derived from other data.

So an obvious question to answer if evaluating a systems ability to fit a given challenge, is: What technical ecosystem is this crowdsourcing platform going to fit into?

4.1 Collection Management System

A Collection Management System, abbreviated CMS (thus sometimes understanably confused with a content management system) is a software system used when managing a collection of historical artefacts. It is typically used by a gallery, library, archive or at a museum. The Collection Management System holds information about each of the artefact in the collections, information that is referred to as metadata. I.e. data about the data.

Besides persistently storing metadata about the artefacts, it also holds information on its provenance and about other events related to a particular artefact: When and how did the institution get the particular artefact into its collection, when and how has the artefact been undergoing restoration and when has it been on loan.

4.2 GenReg

GenReg is the name of a set of Microsoft SQL databases with graphical user interfaces build as Access applications, developed by employees at the museum, and used as a collection management system at the museum.

They are deployed within the physical boundaries of the museum, as they are hosted from the basement of the museum, and as such can only be accessed by employees connected to the local area network of the museum.

A diagram presenting the database model is provided on figure 4, but it is not explicit from this that the GenReg system also contains information about image files. This is not atypical for a collection management system, as such, but the GenReg system has an interesting approach to the association of images to objects, representing artefacts.

An image can be linked to an artefact in one of three ways:

- **Family portrait** – A photograph of multiple related artefacts, possibly found a the same geographical location in ex. a chest.
- **Individual portrait** – A photograph of the single artefact, showing the whole artefact.
- **Detail of individual** – A detailed photograph of the single artefact, showing a particular detail which might be of interest to a curator or visitor at the museum.

A RESTful[4] web service has been implementing to support an integration with the GenReg suite, but the implemenation of the GenReg Mønt extension has not been configured. It is expected that the museum will be migrating away from this system, as it runs only on unsupported Microsoft Windows XP machines with an outdated version of the Microsoft Access Database interface tool. This is probably the reasoning behind the museums fifth sub-goal of migrating into the national system for museums, which is expected to be ready in two years time.

[4] A web service implementing an interface compliant with the definition of a Representational State Transfer (REST) architectural style interface, as first described by Fielding, Roy T. [24, chapter 5]

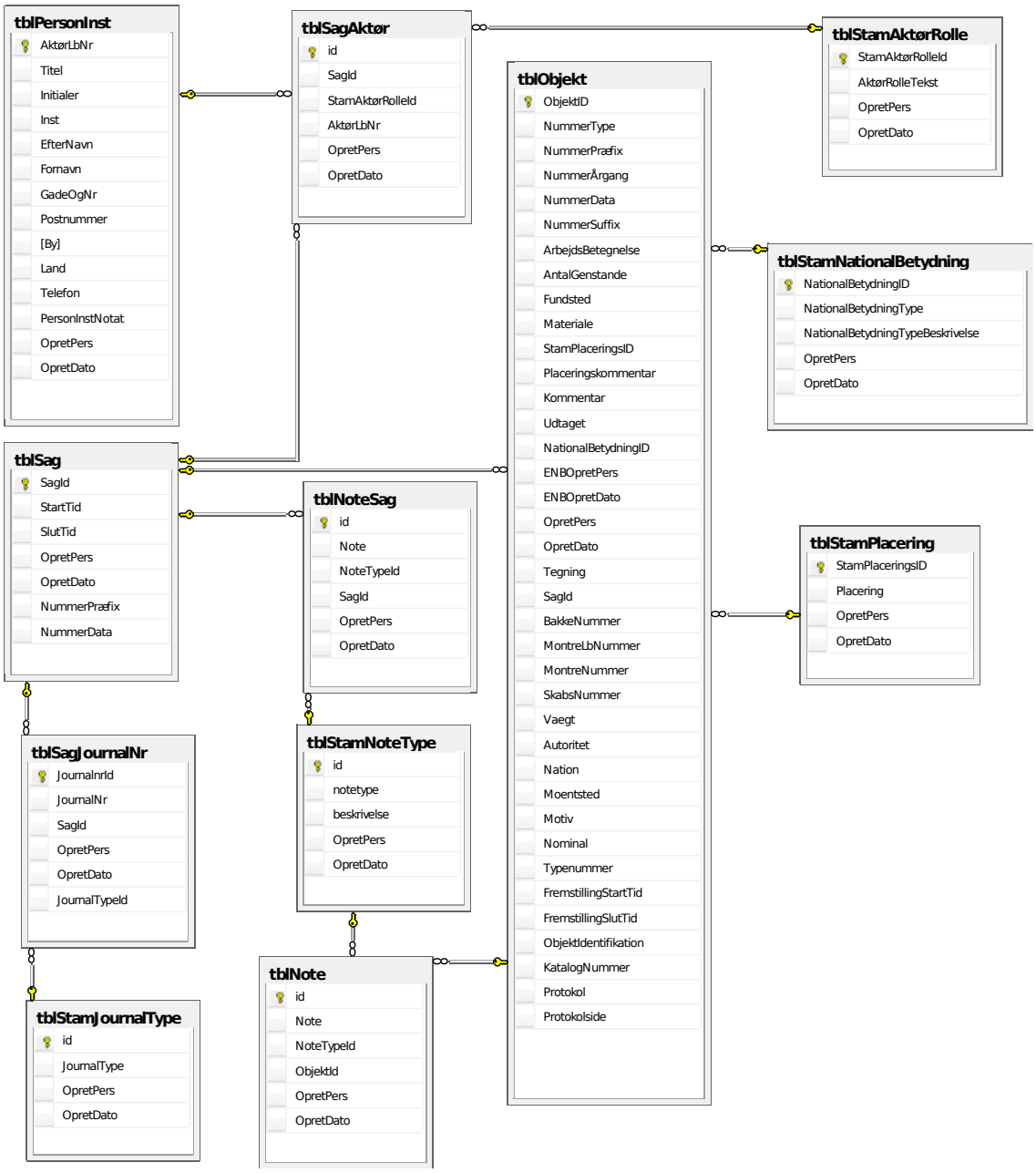


Figure 4: Database model of GenReg mønt, as provided by Bodil Qvistgaard. This dia- gram is not my work but courtesy of Bodil.

4.3 Fælles museums IT

A collection management system, called Regin, is provided to the museums which are recognized by the state of Denmark. It is not a strict requirement that the museums use this system, but it these must report metadata about their collections into a central database, to which Regin is the only known client for reporting.

Regin has inherit problems, one such is a very arbitrary metadata schema – ex the concept of a ship is defined into great detail whereas other types of artefacts are only defined throug an abstract “Object”.

The ministry of culture has initiated a tender, on the development and hosting of a new collection management system to replace Regin, in 2016. Unfortunately the specifications of this system was not released before this thesis was due, but this is definitely an interesting system to keep an eye out for, when planing future functionality of this platform.

4.4 Digital Asset Management

The Digital Asset Management System, abbreviated DAMS, is a system which organizations use to store and categorize their photographs, scans and potentially other media files (such as audio and video clips), for easy retrieval and access across departments or via user-oriented websites and products, such as smartphone applications and alike.

Loading digital media files (often called assets) directly from a shared component in the network, reduces work associated with maintaining the published assets used in a portfolio of products around the museum, but it also helps the curators conform with a shared work flow (curators helping each other), as an asset can be attached metadata referring to its state of digitization. And the activities in relation to digital asset management doesn't have to be influenced by the product it is going to be used in.

From a technical standpoint the use of a DAMS usually increases performance of multiple systems, as previews (i.e. resized representations of a large original image) of images can be shared across multiple end-products. Allowing caching at a single component, instead of a cache at each end-product.

4.5 Canto Cumulus

The National Museum of Denmark uses the Cumulus DAMS, from the German software vendor Canto, hosted through on-premise infrastructure at the museum, by the Danish reseller Attention Solutions.

The Cumulus installation at the museum has an integration component called the canto integration platform, CIP. This component exposes assets to the internet via a web service, an application specific RESTful protocol build on-top the standard hypertext transfer protocol, HTTP.

The features of the CIP are devided into 9 sub services: Session, Metadata, Preview, Asset, Comments, Location, Developer, Configuration, System - of which four has to do with user rights management and configuration and additional two (Comments and Location) will prove to be of little to no interest to this specific usage scenario. [41] Remaining is the following services

Metadata This service has a total of 19 operations. Of which the most relevant is the ability to search for assets using a query language, called the Cumulus Query Format [40], supporting queries returning metadata on assets compliant with a set of conditions which are basically comparisons between values of metadata fields on an the same asset and constant values provided in the query.

Preview This service provides 3 operations, which basically can help a client generate differently scaled and cropped derivatives from the original assets. The service caches the response on the service side, making subsequent requests for the asset in the same dimensions a lot faster than for every client to download the full sized image and generate a downscaled version for thumbnails them-selves.

Asset This service provides 9 operations for managing the most essential entity of the DAMS, the original asset, which is also downloadable through the service.

An asset in cumulus has the following fields, potentially containing metadata:

Throughout this project, the concrete CIP installation at the museum has proven relatively unstable, with frequent breakdowns across multiple hours, requiring a reboot of the servers.

Cumulus Asset Metadata Fields		
Adequate Registration	File Data Size	Latitude
Address	File Format	License
Actor name	Photo number	Location note
Actor note	Photographer	Longitude
Archive group	Creation time - From	Shelf group
Archive name	Creation time - To	Date for retrieval
Archive note	Sufficiently registrated	Retrieval time - From
Archive number	Street	Retrieval time - To
Asset Creation Date	Object number	Original
Asset Modification Date	Horizontal Resolution	OWC-kode
Lighting	Image Height	ZIP-code
Description	Image Width	Record Creation Date
Description note	Inventory number.	Record Modification Date
Image number	IPTC Description	Record Name
City	IV-task	Regional code
Cataloging User	Keywords	Case number
Categories	Classification	Time note
Copyright Notice	Preservation Information	Availability
Date	Map radius	Year
Digital creation data (deprecated)	Map/short titel	

Table 1: A table with all existing metadata fields on a Cumulus Asset.

4.6 How the components are actually used

Even though the existing infrastructure is fulfilling their respective requirements, I see a challenge in the potential confusion coming from the fact that GenReg Mønt contains both metadata and images of coins. The same goes for the asset management system Cumulus, that contains metadata fields such as the city, street and zip-code of the origination of the image as well as the storage location of the artefact on the photograph.

In an ideal world I would like to have seen a clear division of responsibilities between these two components, having a collection management system describing artefacts, all know information about them, and reusing the asset management system to provide images, linked to assets in the collection management system.

4.7 Domain model of the existing system

From interviews with the museum staff and other stakeholders at the museum, an abstract class diagram representing logic entities and their relations within the system as it is. The representation of *asset*s is primarily within the Cumulus DAMS but as mentioned, images on records within the collection management system GenReg Mønt also exists. The *fact*s about artefacts primarily lives in handwritten protocols and in GenReg Mønt as well as some in metadata fields on the records in Cumulus.

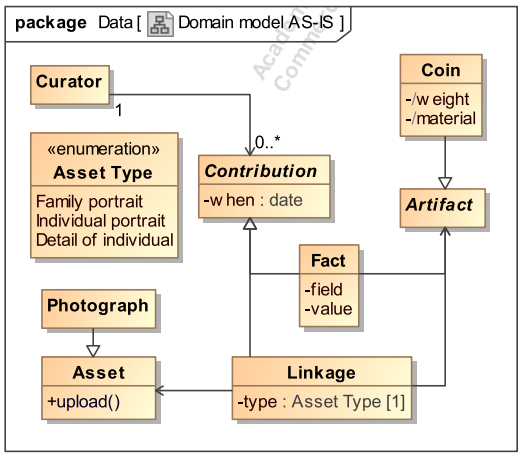


Figure 5: A class diagram presenting the as-is entities as elicited from the product owner.

5 Actors

Throughout my masters program and professional life, I have found that the challenges in an organization that I find the most interesting, tend to involve people.

Really understanding the challenge at hand, involves exploring motives, challenges and constraints of the people which will eventually become users of the product or service. When an organization defines a problem, they tend to have a better understanding of their own needs, than they have an understanding of the context in which the users exists. This is especially the case if the users are external to the organization.

This is why the engineer of the solution must make sure that the description of a challenge is elaborated to include the context of the people, who will eventually become users.

In the attempt of capturing the context of the stakeholders and users, I will introduce and motivate three actors. All of which I have chosen to design the system for, as the system needs to take all three into account if it is going to successfully solve the challenge.

Working with actors simplifies the analysis going onwards, in comparison to having concrete names in design documents. Multiple people might enact the same actor and the same person might enact multiple actors from the systems point of view.

5.1 Curator

An internal employee at the museum responsible for the process of registering artefacts. A curator is responsible for the museums communication with the crowd members on the platform, helping the assurance of the quality of the data produced on the platform. The requirements from the curator is primarily based on qualitative interviews with curator at the museum, Helle Horsnaes, as well as Charlotte S. H. Jensen and Jacob Riddersholm Wang.

The curator has to deal with any physical interaction with artefacts of significant value to the museum, i.e. taking pictures of and weighing the artefacts. This is because coins and medals (which are Helles primary focus) are artefacts of not just cultural and historical value, but some coins and medals can also

have significant economic value, as they are traded and auctioned on the international marked on a daily basis.

This actor also represents the internal IT and database administrators at the museum. It is worth noticing that the number of curators are small, as these are on the payroll of the museum, so their attention is a sparse resource.

As seen from the activity diagram 7 visualising the curators digitisation process. Potential volunteers have practically no tasks to perform in respects to the digitization of precious artefacts.

From a curators point of view, this project can be a little troublesome. In the end it is the curators job to research and contribute facts to the collection and manage the artefacts. The role of the curator is changed from contributing knowledge to verifying knowledge and managing the community with her domain specific knowledge of the artefacts in mind.

In this concrete case some curators has expressed the fear, that the museum would look like it does not have domain experts in the artefacts when it is asking the crowd for help generating data. When implementing a solution this has to be kept in mind, as the narrative used when communicating the project has to address this issue.

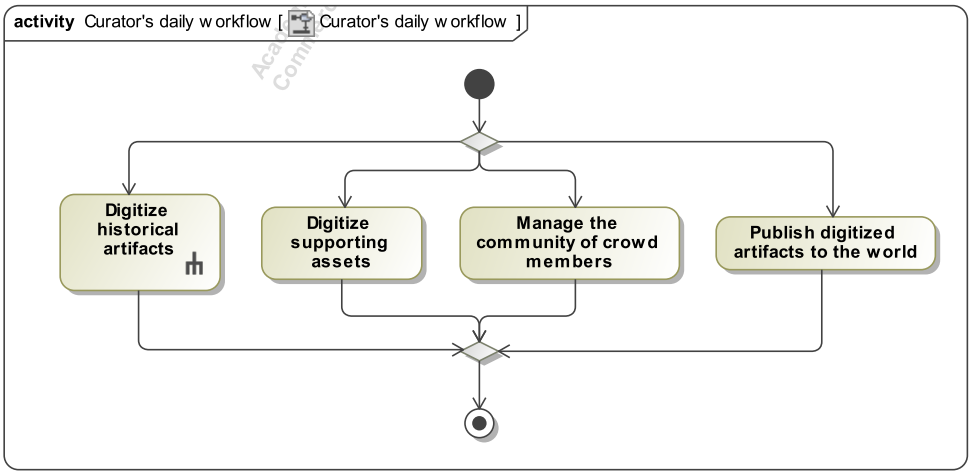


Figure 6: My perception of the daily work flow of the curator with the activities within the scope of the project.

5.2 Member of the Crowd

An external person, interested in the artefacts or in helping the museum or simply looking for a way to spend some time, while possibly feeling entertained or engaged. The set of potential crowd members is fairly large, estimated 5.500 [5].

In conversations with the members of the crowd of Danish numismatics, an issue kept coming up - they are afraid that younger generations would loose interest in their field of research. And they see the digitization of the museums coins and medals as a way to get knowledge about these valuable artefacts out to the public of future numismatics. This became especially relevant after the museum choose to close down the open exhibitions of coins and medals [39].

[5] A quick estimate of people from the head of the Danish Numismatics Association.

Figure 7: My perception of the curator's as-is activity of digitizing historical artefacts at the museum.

The crowd of enthusiasts is important to the museum as they have a lot of domain specific knowledge of information and processes that is normally reserved the museum's curators - which is a sparse resource. At the same time these enthusiasts are loyal users of the museums facilities and most of them have strong connections to the museum as an institution.

Key stakeholders have expressed that it is vital to the success of the museum, that the it keeps a good relation with the crowd of especially numismatics and enthusiasts in general.

Charlote S. H. Jensen imagines the members of crowds incentive to spent time on the platform as dependent on the ability to spend both short and long sessions of time on the playform. It should not require of the members of crowd that they spend a certain amount of hours a week as it should be able to substitute a quick game.

As with any person, the members of crowd has a need for entertainment. The majority of representatives of the actor responded that they didn't play games on a regular basis and one said that he saw it as a waste of valuable time.

Figure 8: A photograph from one of the workshops with representatives from the members of the crowd.

From workshops with the product owner, the following process of contribution from the member of crowd has been elicited. At a workshop each of the activities were tested. As an example the ability to contribute a fact about the artefact was tested by presenting a printed copy of an image of a coin with its label, together a page from the protocol, as the respondent was asked to identify the demonination of the coin.

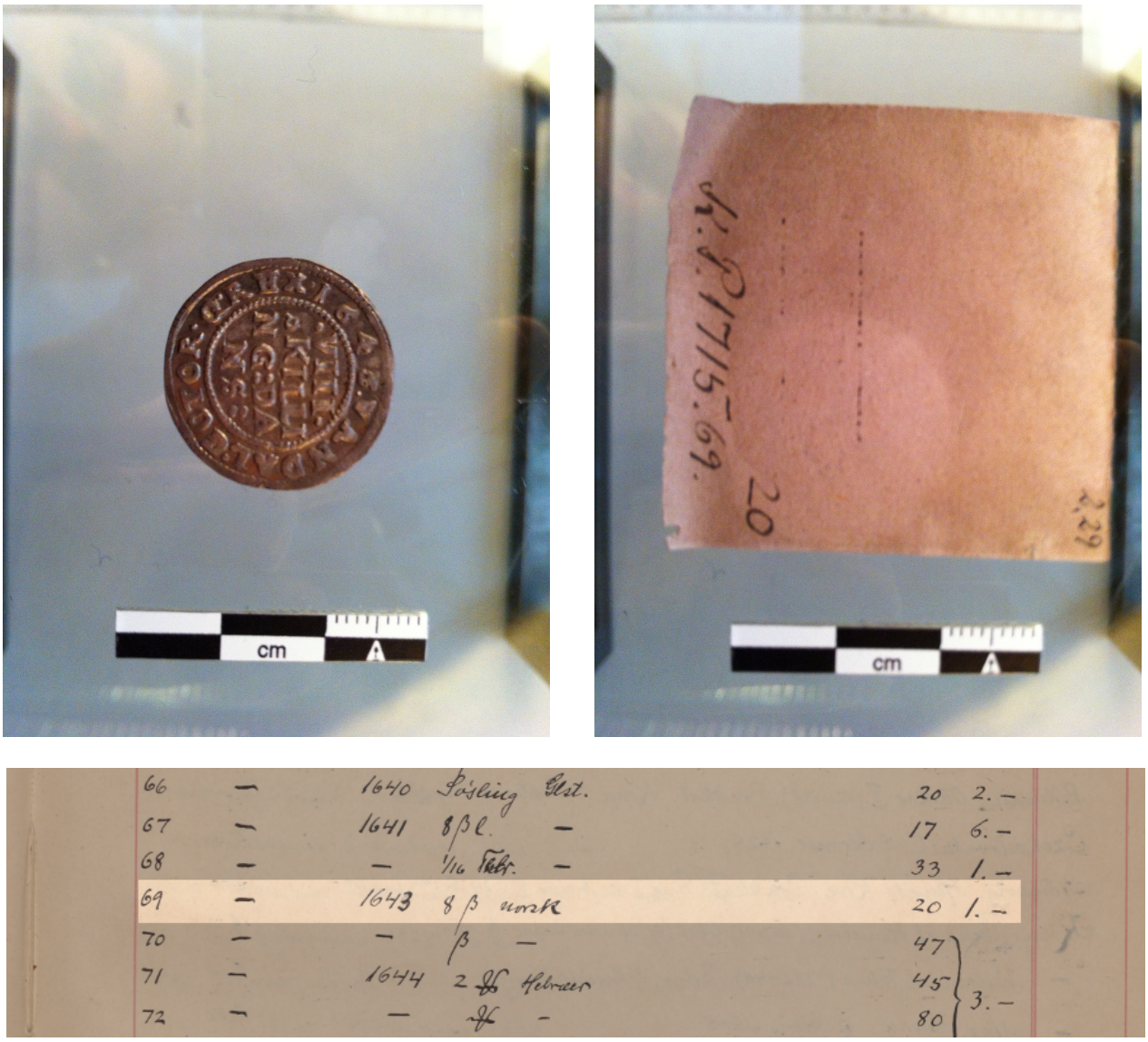


Figure 9: The images used when testing activities with the members of the crowd, the protocol was a whole page without the highlighting provided here.

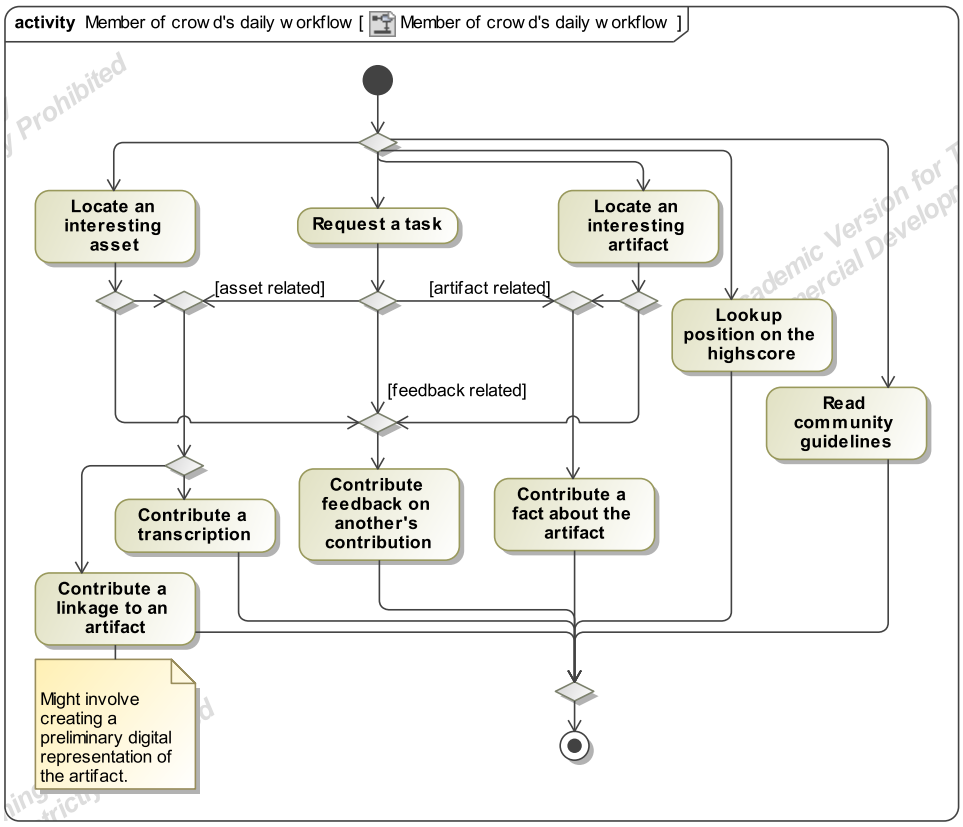


Figure 10: Core business processes to-be of a member of crowd as a result from workshops with stakeholders.

5.3 Deployer

This actor represents the individual setting up the system. Possibly an internal IT administrator at the museum or at some other museum. This actor drives the development for a reusable, easy to set-up and easily maintainable system.

The reason for having this actor represented is because the museum has a strategic attitude to promote reuse of software across the industry of museums, globally. If another museum, with similar challenges, can reuse this generic solution to crowd engagement, it will reflect good on the National Museum of Denmark, ultimately enabling them to deliver into their goal of being recognised as one of the leading museums in digital dissemination.

6 Domain model of the system to-be

From workshops with key stakeholders, such as the product owner, a domain model consisting of entities and their relation of the imagined system to-be deployed. It is also inspired from existing systems for crowd engagement, such as the domain model introduced in the section 3.1 regarding “Politiets Registerblade”.

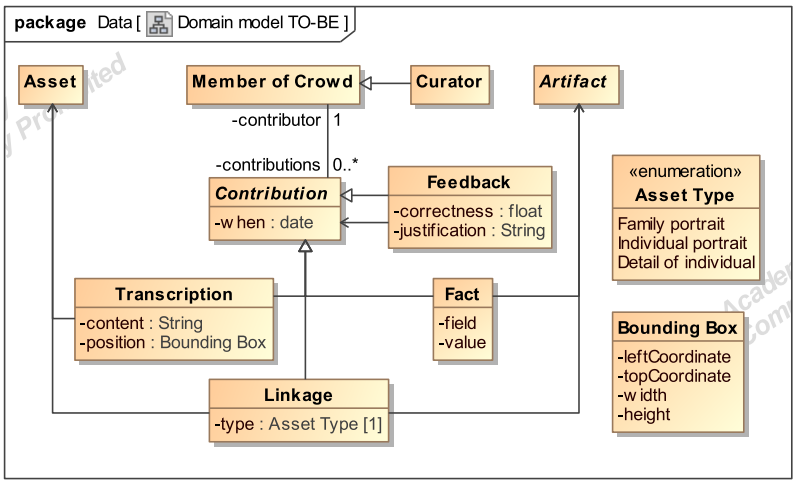


Figure 11: A class diagram presenting the to-be entities as presented by stakeholders.

7 Listing the requirements — the preliminary product backlog

The requirements are gathered under the stakeholder interviews and presented as user stories. They have been ordered by perceived business value (from the product owners point of view). I proposed a preliminary ordering on the basis of the knowledge gathered, later it was approved by the museum's product owner (Jacob Riddersholm Wang).

I introduce each requirement as a user story, with the template advocated by Mike Cohn, amongst others, i.e:

"As a <actor>, I want <some goal> so that <some reason>."[4]

Writing good user stories is not trivial. Bill Wake has introduced the acronym INVEST, to help guide the product owner when defining user stories to be ready for the development team to start designing and implementing. [45] Although the INVEST principle is broadly adopted - I have chosen to deviate from some of the suggestions.

One such deviation is from the suggestion to make user stories small. I would like the preliminary product backlog to contain requirements across the whole product, to make a representative sample for the component analysis. In the attempt to keep the number of stories relatively low, I have chosen to include relatively large stories (some times called epics) as well as non-functional requirements almost so abstract that it could be hard for a developer to understand how to implement it from the textual description. But this might not be needed in this phase - we can refine on the requirements when we know more about the potentially reusable components and as Don Wells mentions, the detailed description might be left for a face-to-face conversation:

"User stories should only provide enough detail to make a reasonably low risk estimate of how long the story will take to implement. When the time comes to implement the story developers will go to the customer and receive a detailed description of the requirements face to face." [47]

The product backlog items presented below are all user stories – this is actually not a requirement, from SCRUM, the items on the product backlog can be anything from user stories, to UML use cases, investigation tasks or even concrete bug-fixes.

I have chosen to present a couple of non-functional requirements on the backlog as well (sometimes called qualitative requirements), which can be interpreted as constraints on how the requirements above that particular non-functional requirement are implemented. [3]

The unique identifier [6] makes it possible to refer to a requirement in a short form across multiple artefacts throughout different phases of the project.

C05 **As a** curator, **I want** to perform any task requiring the physical presence of the artefact (i.e. weighing, photographing, etc.) **so that** the artefacts can be digitally represented in a safe way. [qualitative requirement]

C02 **As a** curator, **I want** any information to carry a trace of meta information on the origin (who / when) and older revisions **so that** the origin of information can be traced. [qualitative requirement]

M02 **As a** member of the crowd, **I want** to contribute an answer (a fact) to a question about an artefact, that I know or can research my way to **so that** I earn points.

[6] Every item on the product backlog starts with an identifier, which consists of a character: The initial letter of the actor from which point of view the requirement is expressed. The identifier also has a two-digit number, which was assigned on an incremental basis in the ideation phase - when writing up the user stories initially - before the product owner introduced a prioritization.

M01 **As a** member of the crowd, **I want** to contribute a transcription of an asset **so that** I or others can find this when searching.

C01 **As a** curator, **I want** information generated by the crowd to carry a disclaimer, that this was not produced by the museums employees, **so that** we cannot be held responsible. [qualitative requirement]

M07 **As a** member of the crowd, **I want** to feel appreciated and as a part of an inner circle, close to the museum **so that** I can keep myself motivated to contribute. [qualitative requirement]

M10 **As a** member of the crowd, **I want** a graphical interface that doesn't "change too much" **so that** I can relax when I contribute. [qualitative requirement]

C12 **As a** curator, **I want** to be able to define and publish community guidelines on how to engage as a member of the crowd **so that** the member of the crowds are guided to contribute the most valuable inputs first.

C08 **As a** curator, **I want** to spend as little time as possible on getting an artefact digitally represented and registered **so that** my time is utilized on valuable manual tasks rather than tasks that could be automated. [qualitative requirement]

C03 **As a** curator, **I want** an overview of recent changes to the information on the system **so that** I can react to the activities of the community.

D06 **As a** deployer, **I want** to configure a subset of the fields from the collection management system, that I want the member of the crowds to contribute information about for a particular type of artefact.

C09 **As a** curator, **I want** to see a list of the most active members of the crowd, **so that** I can contact them in case I want to invite them for an event at the museum.

C11 **As a** curator, **I want** to reward a member of the crowd when a particular artefact has been partly or sufficiently registered **so that** they know we appreciate them and they will remain motivated.

M06 **As a** member of the crowd, **I want** to search for assets from a free-text term or specific metadata fields **so that** I can find the asset relating to a known artefact and eventually link them together.

M13 **As a** member of the crowd, **I want** to search for artefacts from a free-text term or specific metadata fields **so that** I can find the artefact relating to a known asset and eventually link them together.

M05 **As a** member of the crowd, **I want** to be able to link an asset to an artefact **so that** it is easier to find the assets related to the artefact, as they hold information which can be used when contributing facts.

M03 **As a** member of the crowd, **I want** to contribute feedback (verify, decline, flag-as-inappropriate or comment) on a contribution from another member of the crowd **so that** I can earn points.

M09 **As a** member of the crowd, **I want** to have a suggestion on tasks that I can perform **so that** I can get started right away.

C07 **As a** curator, **I want** to reward a member of the crowd when they provide feedback to another member of the crowd **so that** they know we appreciate them and they will remain motivated.

D04 **As a** deployer, **I want** to be able to customize the look and feel of the system, because I want to brand the product with my organisations visual identity **so that** members of the crowd are not in doubt who they are contributing to.

D05 **As a** deployer, **I want** to be able to extend the system with other types of assets and artefacts **so that** we can reuse the set-up in future crowd sourcing projects.

D01 **As a** deployer, **I want** to be able to set-up a simple working demonstration system in less than 15 minutes **so that** I can quickly see if the system solves my challenge. [qualitative requirement]

D02 **As a** deployer, **I want** to be able to integrate the system with a Digital Asset Management System of my choice **so that** we can consolidate our platforms.

D03 **As a** deployer, **I want** to be able to integrate the system with a Collection Management System of my choice **so that** we can consolidate our platforms.

C15 **As a** curator, **I want** the system to automatically fill in known metadata from our collection management system **so that** duplicate work is minimized.

M04 **As a** member of the crowd, **I want** to see the systems perceived correctness of any fact about an artefact **so that** I can see how much I should trust a particular fact and give the contribution feedback or change the fact if I know a better answer.

C04 **As a** curator, **I want** an ability to close down non-constructive debate about a particular artefact **so that** the involved members of the crowd are not demotivating each other.

C06 **As a** curator, **I want** to be able to engage with the system as a member of the crowd as well **so that** I can participate in the registration process as well as get familiar with the member of the crowd's interface if they ask questions about it.

C10 **As a** curator, **I want** to eventually export data to our collection management system once these have been sufficiently verified, **so that** the data can be used elsewhere at the museum.

M11 **As a** member of the crowd, **I want** to suggest additional metadata fields on a particular type of artefact **so that** I can contribute information which would otherwise fall outside the available fields.

M12 **As a** member of the crowd, **I want** to keep my own personal notes on an artefact **so that** I can remember various notes for later use.

C14 **As a** curator, **I want** to have the transcriptions of related assets (protocol pages) available as derived metadata on the artefact **so that** these are available when searching for, viewing or eventually exporting the artefact.

C13 **As a** curator, **I want** to provide suggestions for commonly mistyped words **so that** different ways of typing the word is minimized.

M08 **As a** member of the crowd, **I want** to have assets representing knowledge about artefacts (protocol pages about coins) automatically linked to the particular artefact, whenever the necessary metadata becomes available **so that** I have to do the least amount of work possible to perform a linking of assets to artefacts.

Part II

System fitness analysis

"The most important property of a program is whether it accomplishes the intention of its user."

— Charles Antony Richard Hoare, winner of the 1980 Turing Award

8 Introducing the component analysis

This part of the report deals with the system fitness analysis – a component analysis phase [38, page 35], where a subset of relevant components are analysed to find out how and where they fit in respects to the requirements found in the previous part.

I have chosen a very systematic and mathematical approach on the fitness analysis. One might argue that this is a little too complicated and that the actual measure of fitness might be irrelevant, especially given that the input to the analysis is subjective estimates on an abstract level of reasoning. But as with many processes it is the process itself that brings the value, the conclusion in itself might be expected from the beginning – the analysis brings a language and a focus on potential issues, which would have otherwise remained in the unknown for later discovery. It is expected that the system fitness analysis introduced in this part, would be even more valuable if a large portfolio of reusable components were to be evaluated systematically.

9 Categorizing the requirements

Figure 12: Post-its with user stories while creating the 6 categories.

While I need the requrements for the analysis, evaluating a backlog of 34 requirements against 3-6 candidate systems is too big of an upfront cost. Spending too much time on these initial investigations could be wasted, as they would provide less to no business value in itself, if the project was somehow aborted.

With the intention of simplifying the process of analysing the ability for a subset of systems to comply with the requirements, the requirements are divided in categories, each within an abstract shared topic within the problem domain. The total set of requirements is denoted R (R_c denotes the subset $R_c \subset R$ assigned to the category c).

I have used no special method for producing the categories, except picking user stories off the top of the backlog stack placing each user story in an existing group or creating a new if it didn't seem to fit an existing group.

The following categories appeared:

Community Features supporting the museums management in their encouragement of the community of crowd members. Examples are the crowd feeling being appreciated, a curator being able to detect changes and active members of the crowd while rewarding the members of crowd for their contributions. This also involves customizing the look and feel, moderating debate and curators using the platform as members of the crowd.

Contributing Features supporting the members of crowd when contributing metadata about artefacts and assets, as well as the linking of assets to artefacts. The deployer's ability to select semantic fields for specific types of artefacts and the members of the crowd's ability to have a task suggested as well as being able to propose new semantic fields for a particular type of artefact and keeping personal notes on artefacts.

Effectiveness Features addressing the maximization of effective business processes, respecting the curators need to be the only one in physical contact with the artefacts, the member of crowd's need for a stable user interface and the curator spending as little as possible time on tasks that could be automated or performed by a member of the crowd. The deployer needs a system which is easy to set-up to test if this is something worth using and the curator would like data from the organizations collection management system to be filled in while assets are automatically linked to artefacts.

Integrate Features enabling an easy integration with the museums existing software platform, both supporting alternative types of assets and artefacts from the museums Digital Asset Management System as well as their Collection Management System with a possibility to also export data to the Collection Management System.

Quality Features supporting a maximization of quality of data produced using the system to-be. Making sure the any data produced carries a trace of the origin, as well as a disclaimer. The curator has a possibility to publish community guidelines and the members of crowd can contribute feedback on each others contributions, while the systems perceived correctness of any fact is visible to members of the crowd and suggestions for commonly mistyped words are provided to the members of the crowd.

Retrieval Features supporting the retrieval of data in the system. Supporting free-text as well as structured search on assets and artefacts while having the transcriptions of assets linked to an artefact, directly accessible on the artefact itself.

Category ID c	User stories R_c	# of user stories $\lvert R_c \rvert$
Community	M07, C03, C09, C11, C07, D04, C04, C06	8
Contributing	M02, M01, M05, D06, M09, M11, M12	7
Effectiveness	C05, M10, C08, D01, C15, M08	6
Integrate	D05, D02, D03, C10	4
Quality	C02, C01, C12, M03, M04, C13	6
Retrieval	M06, M13, C14	3

Table 2: User stories in categories.

9.1 Providing a weight for each category

A system should be chosen in a way that delivers the most value to the project's stakeholders first. To respect this we calculate a weight W_c for each of the categories which will later be multiplied onto a subjective valuation of a particular candidate systems ability to fulfil the requirements.

In order to take into account the priority of the requirements on the preliminary backlog, we compute the category weight based on an individual weight w_i contributions from each of the backlog items. Calculated from the rank i of the item on the backlog. $i = 0$ being the most valuable item and $i = N - 1$ being the least valuable item, where $N = 34$ is the total number of items on the preliminary backlog.

It is expected that perceived value decreases as we move down the backlog, but we don't know how fast. We have to assume this if we do not provide the backlog items with an estimated business value. Providing an estimate of business value would probably not do the job in it self, as this could come at a relatively high cost. Problem is, this cost is probably related to the system onto which the feature is implemented, making it difficult to provide a reasonable estimate for the return on investment per. product backlog item.

Therefore I assume a linear decrease of business value across the backlog, defining the weight for a particular item, with rank i as:

$$w_i = \frac{N - i}{N}$$

When defining product backlog items, the level of detail is sometimes hard get just right[7]. Since a candidate system is not know at this point, estimating the workload of designing and implementation a particular feature makes little sense. Some features might have been described in detail through a couple of the user stories while some might be represented in larger user stories.

To address this issue of granularity, it makes sense that the weight of a category is not just the sum but the average of the weights in that particular category.

$$W_c = \frac{1}{|R_c|} \sum_{i=rank(s)|s \in R_c} w_i$$

We want a normalized measure of business value, as we don't care about the absolute value, when we use it as a relative measure. Thus we divide the weight of a particular category with the sum of weights of all categories.

Category ID c	Sum of user story weights $\sum_{i=rank(s)\|s \in C_c} w_i$	Weight of category W_c	Normalized weight of category across all categories
Community	4.2647	0.5331	18%
Contributing	3.8824	0.5546	19%
Effectiveness	3.2941	0.549	19%
Integrate	1.2647	0.3162	11%
Quality	3.5000	0.5833	20%
Retrieval	1.2941	0.4314	15%
Total sum	17.5	2.9676	100%

Table 3: User stories in categories.

[7]The term user story originates from the extreme programming framework, where a granularity where each user story has an estimated workload of 1, 2 or 3 weeks of "ideal development time" is adviced: http://www.extremeprogramming.org/rules/userstories.html

10 How to estimate fitness

By assessing the fitness of a set of different systems, I want to eventually rank a set of candidate systems in respects to their ability to implement with the product backlog on page 23 fastest (in respects to development time).

As discussed in the previous section I have divided the user stories on the backlog in six categories, to save resources on this phase and to enable an abstract discussions on requirements within a shared topic.

The fitness $f_{c,s}$ of a system $s \in S$ in a set of preselected systems S is defined as it's ability to fulfil a specific set c of requirements (i.e. the category c). Later on this could be evaluated through tests with real users.

The measure of fitness of a piece of software to a set of requirements can be hard to determine. An obvious question is what the extreme values are? I.e. what would be the value of fitness for :

- A system implementing a requirement to a satisfactory level?
- A system not implementing the feature at all?
- A system which is implementing a particular feature negatively[8]?

The fitness of the system to a particular feature is the same as the particular features level of implementation within the system.

Another way of maximizing the level of implementation of a particular feature is by optimizing its opposite: To minimize the estimated workload $w_{c,s}$ remaining if the category of features c were to be implemented using an existing system s as the foundation.

Estimating workload can be easier than estimating fitness. It is common practice in SCRUM to estimate the workload of implementing user stories as a key action of the Product Backlog Refinement activity [23, page 13]. One of the benefits is that it forces the developer to ask questions to the product owner about the interpretation of the requirements, which in return lowers the risk of the developer implementing something which is not solving the real problem for the user. Therefore I will determine an estimate of workload and use this as a basis when mapping it to an estimate the fitness.

We can safely define the least fit system in a category c of requirements as the system which requires highest workload to implement, as even worse systems of fitness are not of any interest to the outcome of the analysis, and a system which is more fit then the fittest system within a category is not important, as we assume that S is selected such that it contains the systems which are most likely to fit.

$$\arg\min_{s \in S} f_{c,s} = \arg\max_{s \in S} w_{c,s} \qquad \text{and} \qquad \arg\max_{s \in S} f_{c,s} = \arg\min_{s \in S} w_{c,s}$$

The estimation of product backlog items are often done in story points [23, page 7], a scalar unit expressing the relative workload of tasks (the tasks implicitly referred to by the user story). Story points are not globally translatable to work hours, as they are highly subjective to the team performing the task and their interpretation of an "easy task". Simple methods, such as planning poker[9], exists for estimating the workload of implementing a particular feature in an existing software system.

A system which is fully implementing a category of functionality requires zero story points of workload to fit and because the workload is a linear and open ended scale it is simple to reflect a feature which is double as hard to implement in one system in comparison another system. A negatively implemented feature

[8] Ex. functional implementations or architectural decisions working against a feature or its possible future implementation.
[9] A game played by an agile team when estimating the size of particular user stories.

requires more work to identify, remove and re-implement than simply implementing it from scratch, this is easily represented when estimating workload instead of fitness, as a system with this negatively implemented feature requires more work than implementing the feature without a system.

$$\min_{s \in S} w_{c,s} = 0 \text{ story points}$$

For this analysis the absolute fitness of a system is not of any particular value as it is the relative measure of fitness that we are interested in. Likewise the fitness of systems outside the considered sample S is not particularly relevant to the analysis, so defining the upper bound on the fitness of a system s to a particular category c as 1.0 seems like a reasonable constraint.

$$\min_{s \in S} f_{c,s} = 0.0 \qquad \text{and} \qquad \max_{s \in S} f_{c,s} = 1.0$$

This brings us to the following mapping from workload to fitness of a system s to a particular category (i.e. subset) c of requirements.

$$f_{c,s} = 1 - \frac{w_{c,s}}{\max\limits_{z \in S} w_{c,z}}$$

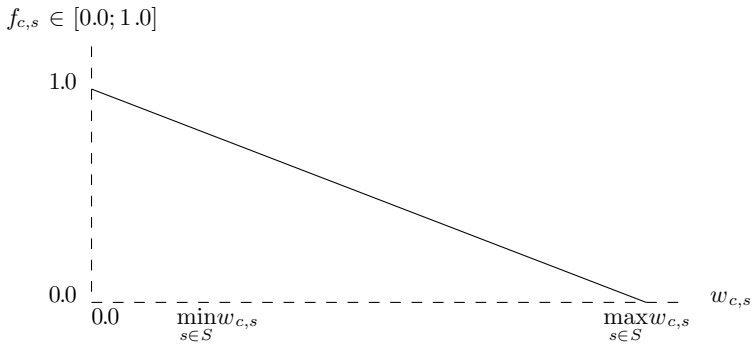


Figure 13: An abstract graph of the workload mapped onto the fitness.

10.1 Estimating workload through planning poker

Any estimation is hard, but a systematic approach to estimation can help making the estimation less biased, by introducing properties to the process making it equally likely to over-estimate as to under-estimate.

Planning poker is one such attempt at making estimation more systematic - and fun.

All participants are given a hand of cards, each card has a number in the Fibonacci-series, from 1 up until 13 (i.e. 1, 2, 3, 5, 8 and 13) from where the numbers are 20, 50 and 100. The number on the card represents an estimated workload in story points - That is, no direct translation between a story points and hours, as this is highly dependent on the team implementing the functionality – but a story of 8 story points is roughly four as large as a story of 2.

The (preliminary) requirements of a software system is understood by the participated parties and in collaboration, the team of estimating developers choose the story representing the least amount of work. This gets the value of 3 story points by definition.

The participants plays one round of planning poker per user story from the top of the backlog – as one of the participants reads aloud the user story and all the participants picks a card from their hand and places it face down on the table. On the end of a 3-2-1 countdown all participants turns around their card and if the story points on the chosen cards are more than three ranks apart (ex. one chooses 1 and another chooses 5), the participants with the extreme values take turn to justify their estimate. The process of picking a card continue until the team converges within three ranks, and the average of the values are picked as the final estimate of the story. This rule of three ranks apart might be changed for a lower number when playing just two people.

Figure 14: A photograph from a planning poker session, with my friend and co-worker Jens Christian Hillerup.

As any approach to estimation, planning poker has the disadvantage of being subjective to the participants. But one of the advantages of playing planing poker is that it provides a less biased approach to estimates than one single person trying to pick estimates from his/her brain. Another obvious advantage is that it forces the development team to ask questions for the product owner and it enables an informal debate on different approaches to solving the challenge of implementing a specific functionality on an abstract level of reasoning.

11 System fitness analysis

In this section I will introduce and evaluate a selected sample (denoted S in the section 10) of software systems for their ability to deliver on the categories of requirements introduced earlier.

The selected sample of systems has been derived from conversations with stakeholders of the project, correspondence with initiators of existing successful crowdsourcing projects mentioned in part 1 of the report and finally the crowd-sourcing projects listed on Wikipedia.org [48].

Because I want to provide requirements that will propose changes to the most fit system I have primarily focussed on systems released under an open source license[10], granting the user the rights to:

- Freely redistribute the software

[10] Read more about the definition of an open source license on http://opensource.org/osd

- Inspect the software's source code
- Produce derived works from the software's source code

I have made this choice to minimize the potential risks of reuse-oriented software engineering, as pointed out by Ian Sommerville [38, Figure 16.2, page 428]:

- *"Increased maintenance costs: If the source code of a reused software system or component is not available, then maintenance costs may be higher because the reused elements of the system may become increasingly incompatible with system changes."*
  This is not applicable for an open source component, as its source code is available from the definition of open source. This is one of the key value propositions when using open source components.
- *"Lack of tool support: Some software tools do not support development with reuse. It may be difficult or impossible to integrate these tools with a component library system. The software process assumed by these tools may not take reuse into account. This is particularly true for tools that support embedded systems engineering, less so for object-oriented development tools."*
  I won't really be using tooling, other than a standard IDE[11]. But with an open source system, a well-documented source code is usually sufficient to enable tool support, such as static analysis tools, debuggers, profilers and alike.
- *"Not-invented-here syndrome: Some software engineers prefer to rewrite components because they believe they can improve on them. This is partly to do with trust and partly to do with the fact that writing original software is seen as more challenging than reusing other people's software."*
  This is something that academia struggles with as well – I argue this is simply a matter of overcoming ones fears of the unknown in the pursuit of the best solution within budget.
- *"Creating, maintaining, and using a component library: Populating a reusable component library and ensuring the software developers can use this library can be expensive. Development processes have to be adapted to ensure that the library is used."*
  Since this component library would be consisting mainly of open source components, the task of maintenance is shared across multiple individuals and organizations.
- *"Finding, understanding, and adapting reusable components: Software components have to be discovered in a library, understood and, sometimes, adapted to work in a new environment. Engineers must be reasonably confident of finding a component in the library before they include a component search as part of their normal development process."*
  By using open source components, the search space is expanded far beyond the organizational boundaries, but the problem of finding suitable open source components remains.

The following three components has been selected for the component fitness analysis:

$$S = \{\text{"MediaWiki"},\ \text{"Amazon Mechanical Turk"},\ \text{"Zooniverse/Scribe"}\}$$

[11] I usually use the open source IDE Eclipse

11.1 MediaWiki

Figure 15: The MediaWiki logo.

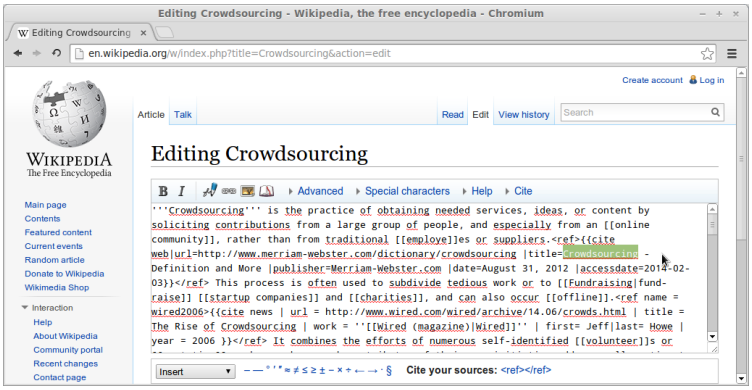


Figure 16: Screen shot of a user editing the WikiPedia "Crowdsourcing" article.

A Wiki is a type of content management system which is designed for quick[12] contribution of content from the same people that is consuming content on the system. The MediaWiki is one such system.

One of the worlds most famous MediaWiki installations is Wikipedia [13] – The Free Encyclopedia, has more than 4.49 million articles contributed to its English version, by volunteers from all over the globe. It is without a doubt the history's most successful crowdsourcing project[14]. MediaWiki is released as Free Software and Open Source software under the GNU General Public License v2.

The MediaWiki is a web-application accessible via the HTTP from a users web-browser. Information is structured in articles about specific topics. Users of the system can create new articles, contribute changes to existing articles and discuss a particular article without making changes to the article itself. The historic trail of changes to an article is preserved in revisions (carrying the change date/time and who contributed the change). An article can refer to one or more media files, such as images, PDFs or alike, which can be uploaded to the MediaWiki. This file handling is implemented through an extendible architecture. User rights management (creating users in different groups with certain permissions across the system) is also handled with extendible architecture.

A developer can change the way the MediaWiki looks and works through extensions. Currently 730 stable extensions has been released for the MediaWiki and 808 is released as beta versions [16][17].

The MediaWiki is a full featured content management system in itself, programmed in PHP[15], HTML, CSS and JavaScript. It is compatible with a couple of SQL compatible databases, such as MySQL 5, SQLite 3+ and PostgreSQL 8.3+, and is able to run on any PHP interpreting web server, such as the Microsoft IIS or the more widely used Apache web server.

The estimation of work below, assumes that the Curator is represented as a MediaWiki Administrator and the Member of Crowd is an Author on the MediaWiki.

[12] Wiki Wiki is Hawaiian for quick.
[13] Available through a web browser on https://www.wikipedia.org/.
[14] Although founder Jimmy Wales does not like to use the term crowdsourcing [32].
[15] The latest version of the MediaWiki (version 1.22) being compatible with version 5.3.2 or above of the PHP language interpreter.

Category ID c and reasoning for the estimation	$w_{c,s}$
Community: A user can see who is contributing what[a]. The logo and overall theme can be changed easily and any administrator is also an author by default. An administrator can protect articles from changes to limit debate on controversial articles. No high score or explicit point system is provided out-the-box to enforce the incentive to contribute, but this might turn out not to be strictly necessary [25, page 3-6] on a MediaWiki or the SocialProfile extension[b] might prove useful for this.	3
Contributing: Articles can be created about artefacts and image files (representing assets) can be upload for later insertion into articles. A user can contribute text to articles and files are automatically attached to an article about them (which can contain a transcription, but editing the article hides the preview of the file). Artefacts could be assigned a type as an article can be assigned one or more categories[c]. Through the Semantic MediaWiki (SMW) extension [12], templates[18] and Semantic Forms[20] the deployer (curator or a member of the crowd) can specify properties which are typically associated with a particular type of article. All articles has a talk article related to it, this can store notes from a member of the crowd, although these will be visible to others. No obvious feature for suggesting a specific task to the contributors of a MediaWiki is known, but one could create lists of articles (using an in-line query[19]) with specific challenges (missing specific values).	8
Effectiveness: Curators will remain the only one in physical contact with artefacts. The use of Semantic Forms can make the user interface more simple and stable across articles of the same type. Articles about artefacts can be automatically created when clicking a link in a list of external data from the collection management system[d]. Pulling data from the CMS into the article at creation time can probably be achieved through the use of pre-loading [14, 8, 9]. If integrated with the museum's asset management system, files can be loaded directly from there instead of being uploaded manually by the curator.	3
Integrate: Image files can be fetched from the asset management system via the CIP as the MediaWiki supports external file back-ends[e], articles about artefacts can be populated with information data from the collection management system by hooking on the 'EditPage::showEditForm:initial'[42] hook and adding content to the article when it is created. Data can easily be imported to and exported from the MediaWiki in various formats (CSV or XML) using the Data Transfer extension[5]. The mapping to the collection management system remains unimplemented.	10
Quality: All changes made to an article carry a date and the author. A disclaimer can be added to the theme if needed, on inserted at the front page (Main article), guidelines can be published as articles themselves. Each revision of an article can be reviewed using the FlaggedRevs Extension[7] but this extension doesn't seem to enable more than one feedback from one person per. article revision, nor does it seem to be able to provide feedback on feedback itself, but this might be provided on the talks page of an article. I would imagine these flags could help to provide a visualization of the systems perceived correctness of an article and therefore the correctness of any fact. It doesn't seem that a solution for the suggestion on replacements for commonly mistyped words is provided as an extension. WikiPedia suggests the users in browser spellchecker for this purpose.[22]	5
Retrieval: All articles can be found using a freetext searchfield in the upper right corner of the MediaWiki, structured search can be performed in the articles with semantic markup, using the SPARQL Protocol and RDF Query Language.[21]	5

[a] The "Special:ActiveUsers page" shows who is active contributors, with a list of all users who has contributed within the last 30 days and how many contributions they've made in this 30 day period. And the "Special:RecentChanges" shows recent changes to articles, either across all users or a particular user.

[b] See as it implements a points and ranking system. [13]

[c] Alternatively name-spaces can be used to categorize and provide specific functionality to articles.

[d] Possible to pull into the MediaWiki using the External Data extension[6], if integrated the RESTful Collection Management System's web service supports a format which is supported by the extension.

[e] Originally created to use other MediaWikies as file back-ends, but build sufficiently abstracted so that you can implement the abstract classes FileBackendStore, FileRepo and UnregisteredLocalFile [15] for the communication with the CIP web service.

Table 4: The estimation of the work remaining in story points for the MediaWiki to fit each category of requirements.

11.2 Amazon Mechanical Turk

Figure 17: The Amazon Mechanical Turk logo.

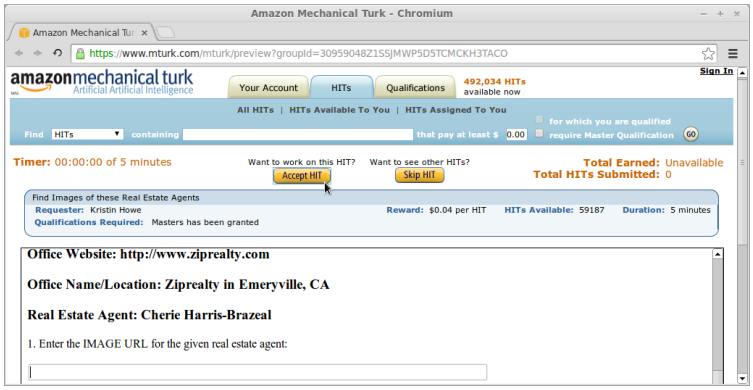


Figure 18: Screen shot of a user answering a HIT regarding finding an image of a particular real estate agent.

The Amazon Mechanical Turk (AMT) is a task-based crowdsourcing platform, allowing individuals or organizations (called Requesters), to propose small tasks for a crowd of workers. The task is called a Human Intelligence Task (HIT), proposing that the tasks that is hard to implement, because it requires human intelligence to complete.

A HIT can be be anything from a simple question with a couple of answers, to a quest for the URL of profile images on a real estate agent's website or a task where the worker is asked to trace the outline of a person in an image. Most HITs are paid a small reward – $0.04 for finding the profile image and $0.05 to trace the outline, but rewards up to $50.00 for providing a translation of a long Portuguese text into English are also available.

To control the quality of the output from the platform, a Requester can reject the completion of a HIT[16] or demand that workers take prerequisites tests before accepting particular types of HITs.

A main reason that I evaluate the fitness of the AMT is because it is mentioned on almost any list of crowdsourcing platforms. It represents a for-profit approach to crowdsourcing and it seems to be a target of studies for academic books, examining the architecture, protocols and algorithms used on a service-oriented crowdsourcing platform [37].

Amazon Mechanical Turk is only legally available for requesters within the USA, and is provided by the Amazon Web Services Inc. (AWS)

The core features of the Amazon Mechanical Turk are:

1. Requesters and workers sign up to the service, using an Amazon Web Services account.
2. Requesters define HITs, as question/options or references to external websites which provide tools to complete the task, such as tracing the outline of a person within an image.
3. Workers accepts and completes a HIT, one at a time, earning money which can be used as gift certificates on amazon.com or paid out (if the worker has a U.S. bank account).

[16] Although this step of the process is not really applicable in this context - as it doesn't scale very well as the requester (probably the museums curator) has to reject completions manually

Amazon Mechanical Turk is not an open source component, but a requester can control it through an web service API, for which AWS provides several software development kits (SDKs), in .NET, Java, Perl, PHP, Ruby or Python, which has all been licensed under open source licenses.[1]

I am asserting that the members of the crowd are workers and the curator is the requester.

Category ID c and reasoning for the estimation	$w_{c,s}$
Community: Given that the members of the crowd are not typical workers (in the AMT context), but selected as people that already have an interest in the artefacts[a], the platform has the inherit problem that it rewards extrinsically through monetary means.[b] When a HIT is accepted and completed, the platform can be queried for the answer – I.e. the changes on facts about artefacts or assets and which workers are active, but an interface for the Curator is not obvious (the Requesters interface is oriented around the management of HITs not about the answers to the HITs). Workers are rewarded in money and can additionally receive a bonus from the requester. As the worker's interface is hosted by Amazon Web Services, little customization of the overall look and feel is possible, although the interface for the HIT can refer to a URL hosted by an external system, which could be branded.	40
Contributing: The workers are not domain experts on artefacts, which is why contributing metadata about an artefact might be hard to realise through the platform, if the member of the crowd cannot be convinced onto it, that being said transcription is a frequent task on AMT. Linking artefacts to a set of assets might be possible on the platform if the artefact had identifiers visible on one or more of the assets, e.g. on a photograph of note or label. The deployer will have to define the semantic fields when building the HIT page from scratch. Suggestions on tasks are the central feature of the AMT platform. As the worker is not a domain expert the proposal of new semantic fields is probably not desirable. As HITs are independent tasks no personal notes are possible nor desirable.	30
Effectiveness: The AMT platform can only handle distributed workers, so no physical contact with artefacts is even possible. The user interface is stable across the same type of HITs, but if the museum has no way to limit what type of HITs the members of the crowd sees, it might change a lot. The AMT can without a doubt help reduce tasks which would otherwise have been performed by the Curator. The system in itself is easy to set-up, but it is hard to scale its benefits without another component serving the HITs to the Workers. Filling in data from the collection management system for verification and automatically linking assets to artefacts is an open problem not addressed with this platform.	20
Integrate: The AMT provides SDKs for the integration with the platform, but this addresses the creation and management of HITs. No obvious solution is provided for the integration with information about assets and artefacts. The answers to HITs can be exported as CSV-files.	10
Quality: All transcriptions will carry an ID of the Worker that did it, as well as the time of submission. No data is presented to the general public, so the disclaimer will have to be added at presentation-time in another context or product. Guidelines can be put into any HIT as an introduction to the task. HITs can be created for peer reviewing, either by asking multiple workers to answer the same question or to verify the correctness of any question with the help of another worker, as a rejection, acceptance or a rating of the initial workers submission. Results are not visible between Workers, so functionality has to be build which can show a facts or transcriptions between workers. Spelling suggestions from the Requester is not a feature of the platform.	10
Retrieval: The answers from the Workers are not indexed for later search, nor is the assets presented on the artefact, as the platform has no features to present artefacts, as such.	15

[a] This assumption could be challenged, but this would require an entirely different approach on the project.
[b] It was an explicit requirement from the members of the crowd, that they don't want to receive money from the museum.

Table 5: The estimation of the work remaining in story points for the Amazon Mechanical Turk to fit each category of requirements.

11.3 Zooniverse / Scribe

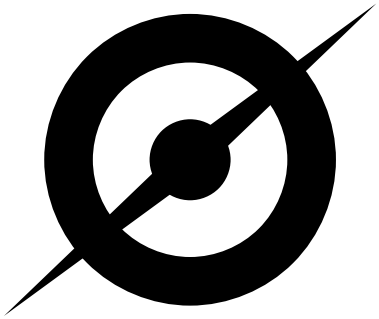


Figure 19: The Zooniverse logo.

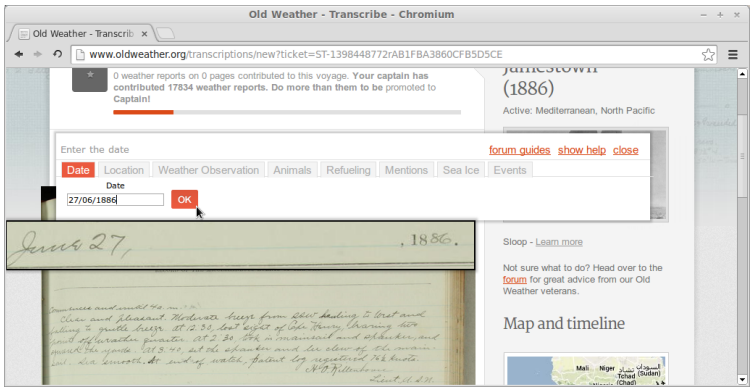


Figure 20: Screen shot of a user providing a transcription of a date using the Zooniverse / Scribe interface.

The Zooniverse is actually not a single product, but rather a portal (with a single sign-on account) allowing volunteering regulars to help scientists and researchers of the Citizen Science Alliance to process data, which is hard to process effectively using computers.

The Zooniverse has been suggested from several independent sources – Charlotte S. H. Jensen, the community manager at the museum and digital archivist Bo Henriksen from Copenhagen City Archives[17]. The project is described on WikiPedia as "Citizen science projects using the efforts and ability of volunteers to help scientists and researchers deal with the flood of data that confronts them".

Many of the Zooniverse parts surround transcription. One such Zooniverse transcription is the "Old Weather" project, asking volunteers to transcribe protocols from ships which sailed the sea between 1850 and 1950 [26]. I have selected this particular sub-project because of its resemblance to the project at the National Museum of Denmark, having transcription of protocols, with entries representing semantic entities.

Many of the technical products used in the Zooniverse project is published under an open source license [35, 50]. An example is the a frontend[18] library for the transcription of hand-written protocols to entities of a data model, called Scribe. Zooniverse describes Scribe simply as "a generalised transcription tool by the Zooniverse".[50, /Scribe]

The core feature of the Zooniverse Scribe is an interface for *transcription* of an *asset*. Each asset belongs to a *collection* of related assets, one such *collection* could represent a book. A *transcription* is performed by a *user* on an *asset*, and it consists of one or more *annotations*, which is transcribed data at a bounding-box on the *asset*. Each asset is associated to a *template*, which is a collection of *entities* that the user can contribute transcriptions of, on the particular *asset*. Each *entity* can represent ex. a date or a specific type of line on the *asset*, such as a weather observation at a particular latitude / longitude. A domain model is provided in figure 21.

The system is implemented in the Ruby programming language, with the jQuery javascript framework and CSS on-top of the Ruby on Rails framework, using a MongoDB document database.

[17] The organisation behind one of the most successful Danish crowdsourcing projects within the cultural heritage industry, politietsregisterblade.dk, mentioned in the requirements section 3.1.

[18] Referring to it actually being interpreted in the end-users web client browser.

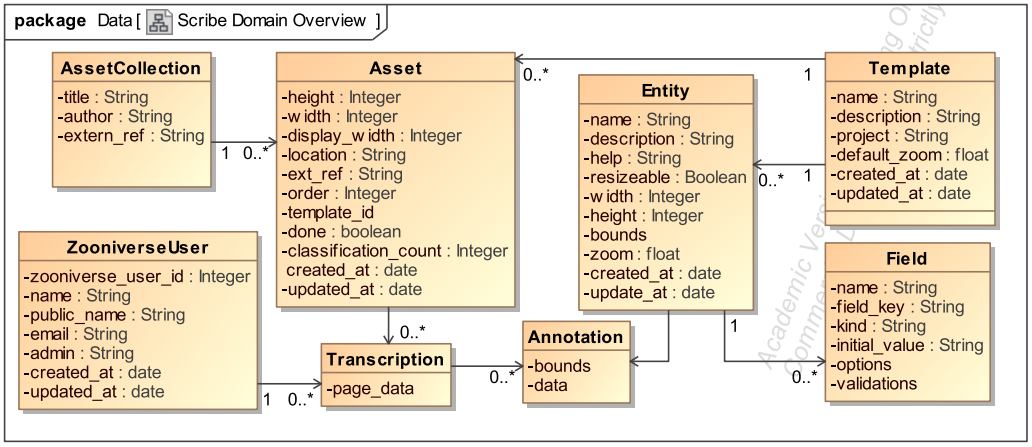


Figure 21: A derived domain model for Scribe.

Category ID c and reasoning for the estimation	$w_{c,s}$
Community: As such Scribe presents almost no contextualization of the task, and no high score is maintained. The curator has no interface to the data, except the direct connection to the database, through a deployer. So implementation is needed to enable the curator to detect changes and active members. No features to reward the members of crowd exists except a congratulating text. The systems look and feel is very basic (and not really visually pleasing) but can be customized via HTML templates and CSS. No debate is possible as members of the crowd cannot see each others transcriptions.	13
Contributing: Scribe is really fit for contribution of transcriptions, but an artefact is only represented implicitly as entities of transcriptions. One could argue that the transcription of an entity representing an artefact represents a linkage of that asset to the abstract/implicit artefact. Deployers create entities which are associated with a template that is associated with an asset at creation time. This way the same template of entities (essentially defining which fields and sematic data is asked for) can be used across multiple assets of the same type. Out of the box the member of crowd gets a list of assets that has not yet had a transcription from the particular user. No feature exists for tasks artefacts as such and the member of crowd cannot suggest new entities for the template. Personal notes could be implemented as an entity or a new field on the transcription model.	5
Effectiveness: The system lets the curator handle any physical contact with artefacts. The user interface is very simple and therefore stable, but the curator has to create any asset in the database via the deployer. The deployer can easily deploy the system, as good guidelines are provided on GitHub. If entities represent artefacts, these are already linked on the asset.	15
Integrate: Nothing is implemented to use the existing software platform. Alternative types of assets are possible at configuration-time, via the creation of templates and controlled when creating the asset of a particular type. The Digital Asset Management System and Collection Management System has to be integrated and data can only be exported directly from the MongoDB by the deployer.	15
Quality: All data created carries the time and person contributing it. Disclaimers can be added to the views and where the data is presented in other systems. The curator can only publish guidelines through the deployer by inserting text into views and members of the crowd cannot see or contribute feedback on other members transcriptions. The system has no indication of perceived correctness of annotations and suggestions for mistyped words is only through the member of crowds web browser.	10
Retrieval: No method of searching in transcriptions are presented. As no view for an implicit artefact exists, assets has no artefact to be linked to, except for the entity data implicitly representing the artefact.	10

Table 6: The estimation of the work remaining in story points for the Zooniverse to fit each category of requirements.

11.4 Calculating fitness

First we calculate the maximal value of estimated workload in story points, for each of the six categories.

c	$\max_{z \in S} w_{c,z}$
Community	40
Contributing	30
Effectiveness	20
Integrate	20
Quality	10
Retrieval	15

Table 7: A table representing the fitness $f_{c,s}$ of the three candidate systems s to each of the categories c.

Transferring the estimated workload from the tables 4, 5 and 6, the estimated fitness of each system to the category of requirements are calculated.

Category c / System s	Community	Contributing	Effectiveness	Integrate	Quality	Retrieval
	Estimated workload $w_{c,s}$					
MediaWiki	3	8	3	10	5	5
Amazon Mechanical Turk	40	30	20	10	10	15
Zooniverse / Scribe	13	5	15	15	10	10

	Estimated fitness $f_{c,s}$						Weighted average of fitness f_s
MediaWiki	0,93	0,73	0,85	0,33	0,50	0,67	69,2%
Amazon Mechanical Turk	0,00	0,00	0,00	0,33	0,00	0,00	3,5%
Zooniverse / Scribe	0,68	0,83	0,25	0,00	0,00	0,33	37,2%

Table 8: A table representing the fitness $f_{c,s}$ of the three candidate systems s to each of the categories c.

This reveals that according to this particular component fitness analysis the MediaWiki is the fittest of the proposed systems with a weighted score of $69,2\%$, making it a great base-component for reuse compared to the other two systems. The Amazon Mechanical Turk gets the worst score of $3,5\%$ - that is not to say that this system is not good for anything, it is just not fitter than the worst known system to the analysis. If the analysis included an estimate of workload for the implementation of the requirements without a base component system, the fitness of the AMT would increase, as this is now compared to something which is even worse, but this doesn't change the fact that the MediaWiki is the obvious choice as a component to develop around.

Another interesting observation is that the Scribe system seems to be more fit than the MediaWiki in the category of "Contributing", this is primarily because of it's ability to transcribe assets, which a design or implementation onto a MediaWiki could probably be heavily inspired from or partly reused. Some of the frontend JavaScript code from Scribe could probably be reused to provide a transcription interface for files on the MediaWiki.

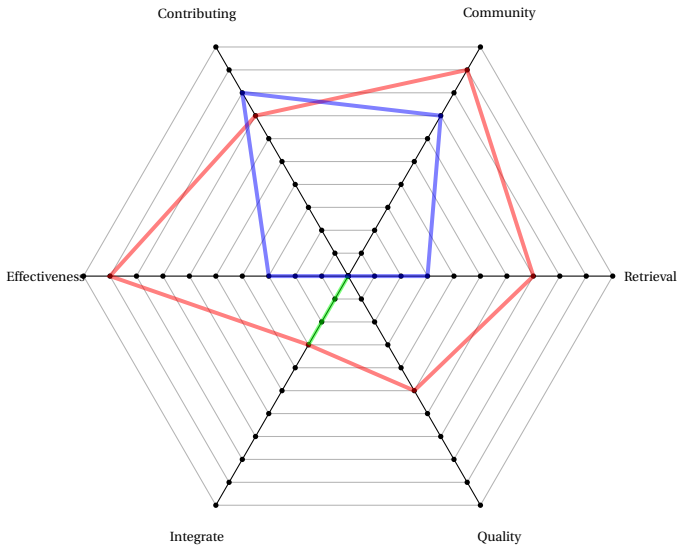


Figure 22: Radar chart visualizing the fitness of MediaWiki (red), Amazon Mechanical Turk (green) and Zooniverse / Scribe (blue)

The reasoning behind the workload estimates for the Amazon Mechanical Turk also reflects a mismatch in actors across the different system. For one thing the AMT is designed for non-domain experts, which reflects badly in its features for providing community management.

12 Requirements modification

With knowledge on the fitness of a couple of proposed components, it is now time to modify the requirements to make the product backlog of user stories, more INVEST [45] and reflect the situation that we now know how the components play together.

As an overall requirement of the project going forward, it is safe to assume that a product based on the MediaWiki is reasonably good idea. This entails that new functionality is provided through the installation or design and implementation of extensions, as this is the MediaWiki's approach to extending functionality within the framework.

The Zooniverse's Scribe might be introduced by its implementation of a user interface for providing annotations on images.

When choosing the MediaWiki, one obvious compromise is that the system was not originally designed for the members of crowd, but for an actor the wiki calls a *user*. A specialization of the MediaWiki *user* is the *administrator*, which would fit fairly well with the Curator of this project's requirements. One could choose to adapt these names for the MediaWikies actors, but this would risk steering the design off into a direction that might not be targeted towards the actual member of the crowd. The names on the actors keep a focus on the needs in the situation of the real people using the system in the end.

12.1 Elaborated stories

M02 **As a** member of the crowd, **I want** to contribute an answer (a fact) to a question about an artefact, that I know or can research my way to **so that** I earn points.

This is elaborated into **M02a** that includes the semantic field and article entity of the MediaWiki.

M01 **As a** member of the crowd, **I want** to contribute a transcription of an asset **so that** I or others can find this when searching.

This is elaborated into **M01a** to include the file, which can represent an asset, if uploaded to the MediaWiki.

C01 **As a** curator, **I want** information generated by the crowd to carry a disclaimer, that this was not produced by the museums employees, **so that** we cannot be held responsible. [qualitative requirement]

This is elaborated into **C01a** to give a more concrete interpretation of it, specifying the location of this disclaimer.

M07 **As a** member of the crowd, **I want** to feel appreciated and as a part of an inner circle, close to the museum **so that** I can keep myself motivated to contribute. [qualitative requirement]

This is elaborated into **M07a** to acknowledge that the MediaWiki motivates through the intrinsic motivation of contributing for a purpose of greater good - this is why the WikiPedia continues to recruit new contributors and this is why the framing of the mutual benefits in the “Politiets registerblade” project, section 3.1, was so important.

M10 **As a** member of the crowd, **I want** a graphical interface that doesn't "change too much" **so that** I can relax when I contribute. [qualitative requirement]

This is elaborated into **M10a** to capture a more precise formulation of the same requirement, and **M10b** to capture the need of translation of all features.

C12 **As a** curator, **I want** to be able to define and publish community guidelines on how to engage as a member of the crowd **so that** the member of the crowds are guided to contribute the most valuable inputs first.

This is elaborated into **C12a** to give a more concrete interpretation – this is concrete enough that it is almost a trivial task.

C08 **As a** curator, **I want** to spend as little time as possible on getting an artefact digitally represented and registered **so that** my time is utilized on valuable manual tasks rather than tasks that could be automated. [qualitative requirement]

This requirement is actually a pseudo requirement, already captured in the fact that we have an actor representing individuals outside the organisation. This is elaborated into **C08a** eliminating the need for the curator to upload image files manually for the members of crowd to transcribe (**M01a**).

D06 **As a** deployer, **I want** to configure a subset of the fields from the collection management system, that I want the member of the crowds to contribute information about for a particular type of artefact.

This is elaborated into **D06a** to give a more concrete interpretation – including the semantic fields and the use of semantic templates.

C11 **As a** curator, **I want** to reward a member of the crowd when a particular artefact has been partly and sufficiently registered **so that** they know we appreciate them and they will remain motivated.

This is elaborated into **C11a** and **C11b** to focus on the reward of points as introduced by the SocialProfile extension [13], as well as articles about artefacts. **C11a** focusses on the action of editing and article where **C11b** focusses on an edit that introduces values to all semantic fields representing an adequate registration.

M06 **As a** member of the crowd, **I want** to search for assets from a free-text term or specific metadata fields **so that** I can find the asset relating to a known artefact and eventually link them together.

This is elaborated into **M06a** to give a more concrete interpretation – including the semantic values and articles.

M13 **As a** member of the crowd, **I want** to search for artefacts from a free-text term or specific metadata fields **so that** I can find the artefact relating to a known asset and eventually link them together.

This is elaborated into **M13a** to give a more concrete interpretation – including the semantic values and articles.

M03 **As a** member of the crowd, **I want** contribute feedback (verify, decline, flag-as-inappropriate or comment) on a contribution from another member of the crowd **so that** I can earn points.

This is elaborated into **M03a** to give a more concrete interpretation of the different types of feedback.

From the original to-be domain model 11 a contribution is either a transcription, fact, linkage or a feedback itself and the type of feedback envisioned was a one-dimensional measure of correctness with a text string stating the justification. The user story provides different interpretations of this one-dimensional correctness, essentially proposing meaning for different values of the correctness:

- verification – this is a high correctness feedback
- rejection (to decline a change) – this is a low correctness feedback
- flag-as-inappropriate – this might actually be a variant of the rejection, with a justification focusing on the presence of irrelevant information rather than incorrect.

Finally the to-be domain model provided the justification attribute of the Feedback class as a means of commenting while providing feedback.

The component analysis revealed that the MediaWiki could be extended with functionality to flag revisions with the FlaggedRevs Extension [7] - to provide feedback on each change of an article. A flag is an evaluation of a revision (i.e. state) of an article in respects to a set of tags (i.e. dimensions of evaluation), these tags are configurable – the extension's documentation suggests Accuracy, Depth and Tone in each three levels [7, Basic settings]:

Tag names	Level names		
Accuracy	Low	Medium	High
Depth	Superficial	Sufficient	Detailed
Tone	Weak	Good	Excellent

Table 9: FlaggedRevs Extension proposed flag-tags.

This can provide an ability to provide feedback on the contribution of a transcription (a change of an article about an asset), fact (a change of an article about an artefact), linkage (the change of an article about an artefact where a file is inserted for the first time).

One of the short comings of this approach of interpretation of the requirements, is that users cannot provide feedback on feedback itself. Another potential issue with this approach is that what is evaluated is the article after the contribution, not really the contribution itself. This is - if a good article is changed for the worse, the new revision can still be flagged as highly accurate, detailed and in a good tone – although the actual change might have changed the tone from excellent. It is up to the product owner to decide if this is an okay compromise to make in the trade-off for a bunch of functionality essentially solving all other aspects of feedback.

One could extend the FlaggedRevs extension to implement feedback on feedback later - so leaving it out of the requirements might be okay for the first iteration. Feedback on feedback on feedback on a fact about an article is actually a conversation about the content of the article - this is provided out of the box by the talks page associated to any article.

M09 **As a** member of the crowd, **I want** to have a suggestion on tasks that I can perform **so that** I can get started right away.

An actual entity representing a task in the domain model is not be provided by the MediaWiki – but an article describing how to find assets in need of a transcription and artefacts in need of semantic values or articles in general in need of feedback, might be a valid solution.

This is elaborated into **M09a** to give a more concrete interpretation – using references to special pages and the use of a guiding article.

C07 **As a** curator, **I want** to reward a member of the crowd when they provide feedback to another member of the crowd **so that** they know we appreciate them and they will remain motivated.

This is elaborated into **C07a** to give a more concrete interpretation – using the term points from the SocialProfile extension as well as the flag form the FlaggedRevs Extension.

D04 **As a** deployer, **I want** to be able to customize the look and feel of the system, because I want to brand the product with my organisations visual identity **so that** member of the crowds are not in doubt who they are contributing to.

This is elaborated into **D04a** to give a more concrete interpretation – requiring the museum's concrete visual identity to be implemented in the product.

D01 **As a** deployer, **I want** to be able to set-up a simple working demonstration system in less than 15 minutes **so that** I can quickly see if the system solves my challenge. [qualitative requirement]

The MediaWiki provides this ability as such, but this requirement is interpreted into **D01a** – a need for an installation guide.

C15 **As a** curator, **I want** the system to automatically fill in known metadata from our collection management system **so that** duplicate work is minimized.

This is elaborated into **C15a** to give a more concrete interpretation – mentioning articles and semantic fields as well as the prerequisites for this functionality.

C10 **As a** curator, **I want** to eventually export data to our collection management system once these have been sufficiently verified, **so that** the data can be used elsewhere at the museum.

This is elaborated into **C10a** to give a more concrete interpretation – mentioning articles and semantic fields.

C14 **As a** curator, **I want** to have the transcriptions of related assets (protocol pages) available as derived metadata on the artefact **so that** these are available when searching for, viewing or eventually exporting the artefact.

This is elaborated into **C14a** to give a more concrete interpretation – mentioning articles and semantic fields.

C13a **As a** curator, **I want** to provide suggestions for commonly mistyped words **so that** different ways of typing the word is minimized.

This is elaborated into **C13a** to give a guide on how to use the browsers build in spell checking as well as a list of commonly mistyped words and language guidelines.

M08 **As a** member of the crowd, **I want** to have assets representing knowledge about artefacts (protocol pages about coins) automatically linked to the particular artefact, whenever the necessary metadata becomes available **so that** I have to do the least amount of work possible to perform a linking of assets to artefacts.

This is elaborated into **M08a** to give a more concrete interpretation – mentioning articles.

12.2 Eliminated stories

C05 **As a** curator, **I want** to perform any task requiring the physical presence of the artefact (i.e. weighing, photographing, etc.) **so that** the artefacts can be digitally represented in a safe way. [qualitative requirement]

This is the premise of the business processes and the MediaWiki respects this.

C02 **As a** curator, **I want** any information to carry a trace of meta information on the origin (who / when) and older revisions **so that** the origin of information can be traced. [qualitative requirement]

This could be elaborated into a story which captures the logging of change to articles instead of the abstract term “meta information”, as this is a standard feature of the MediaWiki, this is eliminated.

C03 **As a** curator, **I want** an overview of recent changes to the information on the system **so that** I can react to the activities of the community.

This is achievable through the RecentChanges special page, which is a standard feature of the MediaWiki.

C09 **As a** curator, **I want** to see a list of the most active members of the crowd, **so that** I can contact them in case I want to invite them for an event at the museum.

This is achievable through the ActiveUsers special page, which is a standard feature of the MediaWiki.

M05 **As a** member of the crowd, **I want** to be able to link an asset to an artefact **so that** it is easier to find the assets related to the artefact, as they hold information which can be used when contributing facts.

This is achievable by inserting files into articles about artefacts.

D05 **As a** deployer, **I want** to be able to extend the system with other types of assets and artefacts **so that** we can reuse the set-up in future crowd sourcing projects.

This is already a feature of the MediaWiki as the type of the asset or artefact is typically implemented as a category or name-space[19] on the article describing either.

M04 **As a** member of the crowd, **I want** to see the systems perceived correctness of any fact about an artefact **so that** I can see how much I should trust a particular fact and give the contribution feedback or change the fact if I know a better answer.

This is realized when the modified requirement **M03a** is implemented - rendering this user story obsolete.

C04 **As a** curator, **I want** an ability to close down non-constructive debate about a particular artefact **so that** the involved member of the crowds are not demotivating each other.

If a user has the *sysop* permissions, it can protect an article from further editing - so this is a feature of the Media Wiki out of the box.

[19] A MediaWiki namespace is a prefix to the articles title, seperating the title of the name-space with the title of the article by a colon.

C06 **As a** curator, **I want** to be able to engage with the system as a member of the crowd as well **so that** I can participate in the registration process as well, as get familiarised with the member of the crowds interface if they ask questions about it.

Using the MediaWikies user groups this is a feature provided out of the box.

M11 **As a** member of the crowd, **I want** to suggest additional metadata fields on a particular type of artefact **so that** I can contribute information which would otherwise fall outside the available fields.

The elaborated story **D06a** makes this possible for any user - the curator could project the template article if this is suddenly not desired any more.

M12 **As a** member of the crowd, **I want** to keep my own personal notes on an artefact **so that** I can remember various notes for later use.

This is possible to some extend, using the talk articles related to the particular member of crowd's user or the artefact in consideration. Although this is not private notes - this might be an issue that the product owner would object against, in this case the privacy should be emphasized in a alternative story.

12.3 A modified backlog

Below is the modified backlog - incorporating the results on the component analysis.

M02a **As a** member of the crowd, **I want** to contribute a value for a semantic field related to an article about an artefact, that I know or can research my way to **so that** I earn points.

M01a **As a** member of the crowd, **I want** to contribute an article as a transcription of an image file representing an asset **so that** I or others can find this when searching.

C01a **As a** curator, **I want** every article to carry a disclaimer in the footer of the web page, telling visitors that "The content on this website has not been manually verified by the museum and it's employees." **so that** we cannot be held responsible.

M07a **As a** member of the crowd, **I want** to have the purpose of my contributions articulated on the frontpage of the website **so that** I can keep myself motivated to contribute.

M10a **As a** member of the crowd, **I want** a graphical interface with a consistent look and feel across the majority of functionalities **so that** I can relax when I contribute. [qualitative requirement]

M10b **As a** member of the crowd, **I want** the majority of text presented to be formulated in the same language and in a language I understand - preferably my native language **so that** I can relax when I contribute. [qualitative requirement]

C12a **As a** curator, **I want** an article that describes community guidelines on how to engage as a member of the crowd **so that** the member of the crowds are guided to contribute the most valuable inputs first.

C08a **As a** curator, **I want** assets to be loaded directly from the asset management system when referenced **so that** my time is utilized on valuable manual tasks rather than tasks that could be automated.

D06a **As a** deployer, **I want** to create a template article with a configuration of the subset of semantic fields from the collection management system, that I want the member of the crowds to contribute information about for a particular type of artefact**, so that** they are guided to contribute the values that we thing we will need elsewhere.

C11a **As a** curator, **I want** to reward a member of the crowd with points when an article about an artefact has been edited **so that** they know we appreciate them and they will remain motivated.

C11b **As a** curator, **I want** to reward a member of the crowd with points when an article about an artefact has been edited to have all semantic fields filled with values **so that** they know we appreciate them and they will remain motivated.

M06a **As a** member of the crowd, **I want** to search for file-articles about assets from a free-text term or specific semantic value associated to the article **so that** I can find the asset relating to a known artefact and eventually link them together.

M13a **As a** member of the crowd, **I want** to search for articles about artefacts from a free-text term or specific semantic value associated the the article **so that** I can find the artefact relating to a known asset and eventually link them together.

M03a **As a** member of the crowd, **I want** to contribute a flag on a revision of an article, providing an evaluation of the correctness / accuracy of the current state after the change to the article **so that** I can earn points.

M09a **As a** member of the crowd, **I want** to have an article (easily accessible) with lists of assets in need of transcriptions, artefacts in need of semantic values and articles in need of feedback **so that** I can get started right away.

C07a **As a** curator, **I want** to reward points to a member of the crowd when they provide a flag on a revision of an article by another member of the crowd **so that** they know we appreciate them and they will remain motivated.

D04a **As a** deployer, **I want** a customized look and feel of the system, in accordance with the visual identity of the museum (including logo, colours and fonts) **so that** member of the crowds are not in doubt who they are contributing to.

D01a **As a** deployer, **I want** a guide providing instructions on how to install/deploy the system in less than 15 minutes **so that** I can quickly see if the system solves my challenge.

D02 **As a** deployer, **I want** to be able to integrate the system with a Digital Asset Management System of my choice **so that** we can consolidate our platforms.

D03 **As a** deployer, **I want** to be able to integrate the system with a Collection Management System of my choice **so that** we can consolidate our platforms.

C15a **As a** curator, **I want** the system to automatically fill in known values of the semantic fields from our collection management system when an article about an artefact is created **so that** duplicate work is minimized.

C10a **As a** curator, **I want** to export articles and their semantic values to our collection management system once these have been sufficiently verified through flags, **so that** the data can be used elsewhere at the museum.

C14a **As a** curator, **I want** to have the content of file-articles linked (essentially transcriptions of assets – such as protocol pages) available as derived semantic fields on the artefacts article **so that** these are available when searching for, viewing or eventually exporting the artefact.

C13a **As a** curator, **I want** to provide guidelines on how to use the browsers build in spell checking as well as a list of commonly mistyped words and general language guidelines **so that** the amount of different ways of typing the same information is minimized, to make searching easier and metadata less ambiguous.

M08a **As a** member of the crowd, **I want** to have file-articles representing assets with knowledge about artefacts (protocol pages about coins) automatically linked to the particular artefact's article, whenever the necessary semantic values becomes available **so that** I have to do the least amount of work possible to perform a linking of file-articles with assets to artefact articles.

Part III

Designing and implementing a solution

"Talk is cheap. Show me the code."

— Linus Torvalds, primary author of the Linux operating system

13 Incrementally designing and implementing a solution

A candidate system has been proposed, forming a platform onto which the modified product backlog of functionality can be designed and implemented. In order to assure working functionality, each of the user stories are implemented one at a time, in the order of perceived business value, only progressing to the next story when the first has been implemented and documented in the report.

The implemented code is referenced in appendix A.

This phase of the project is essentially a synthesis new and modification of existing functionality.

It is not expected that all of the backlog is designed and implemented within this sprint, and as this is the initial sprint no data on the expected velocity exists. The user stories have not been estimated individually either, so I will simply be trying to implement as much as possible before the project deadline.

13.1 M02a

This is implemented with an installation of the Semantic Media Wiki [12] and Semantic Forms [11] extensions – this can be done simply by adding it to the dependency management file `composer.json`

For every type of artefact - to start with, coins and medals - a class is created in the Semantic MediaWiki following by navigating to the CreateClass special page and entering the properties.

When navigating to the “Start of form” special page, the user gets an input field requesting the name of a new or existing article as well as the selection of the form to use when contributing information. What information the user is asked of is not considered a part of this particular user story, as this is an expected result of implementing D06a. When the user has navigated the article about a particular artefact, a link is presented to “Edit [the article] with [a] form”.

Beneath the fields of the form, a free text field is provided for any non-structured wiki text about the artefact.

See appendix B for screenshots of this implementation.

13.2 M01a

When navigating to an image-file on the MediaWiki one can automatically create an article about the image-file, the only problem being that the image disappears and a text field is introduced when the user has to enter content for the new article.

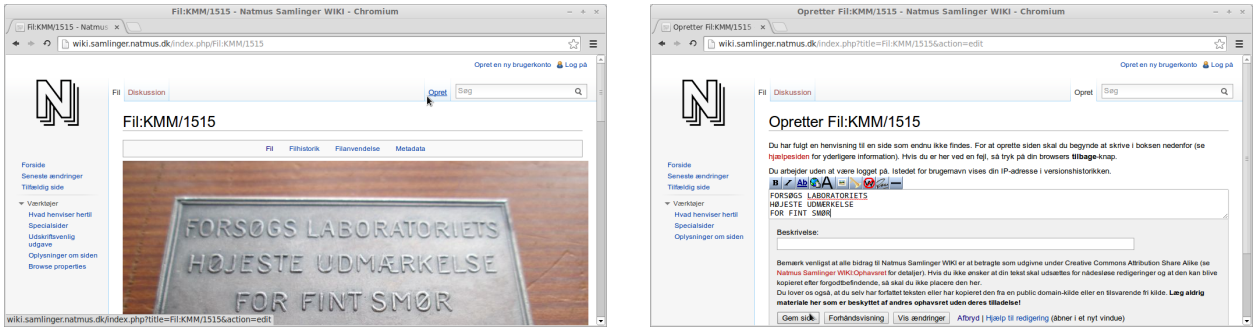


Figure 23: A screenshot of an article about a file before and meanwhile it is created.

The Proofread Page extension [10] essentially provides the functionality requested, and it is being actively used on wikisource.org for transcription of books.

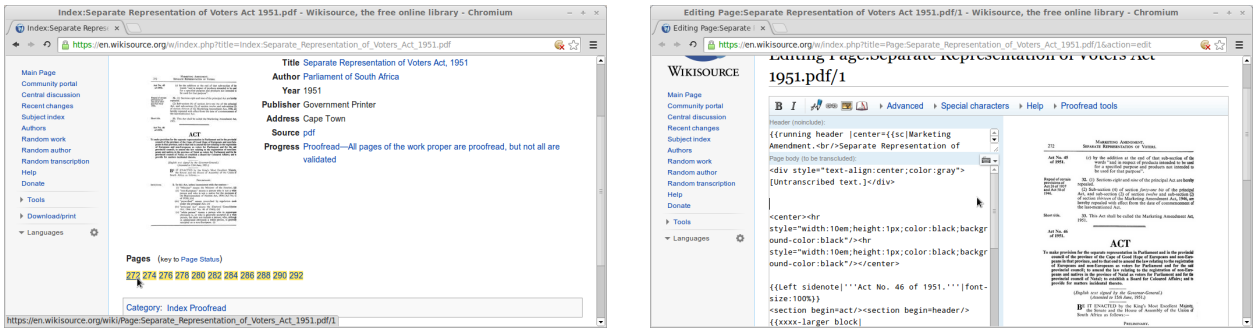


Figure 24: A screenshot of an Proofread index of a PDF and the Proofread interface for transcription on wikisource.org

I tested this interface with the members of crowd, and they found it both counter intuitive and messy, eventually leading to the **M10a** requirement.

When looking closer into the Zooniverse frontend library it was too tightly coupled with the API that the Ruby-on-Rails project provided – so I decided to look into how the frontend library was structured and I ended up using the jquery.imgareaselect library instead.

I've chosen to implement this using the frontend library used in the Zooniverse Scribe (called imgAreaSelect [49]) by creating a new extension to the MediaWiki, called "Transcribe", that:

1. Registers a new special page, titled "Transcribe".
2. Loads a preview of the image file (provided as a tailing argument to the URL) into the special page, as big as possible.
3. A semantic ask query is performed to fetch all values of a specific property (TranscripedContent) in the files article.
   (a) Each of the TranscripedContent values are parsed and added to a table to the right of the image preview.
4. When one of these TranscripedContent values are clicked an area on the preview image is selected and the textual content is shown (editable) below the selection.
5. Holding the mouse over one of these will reveal where they are located at the preview image.
6. Clicking anywhere outside an outline on the preview image will create a new TranscribedContent property.
7. The user can click a save button that saves the modified TranscribedContent values in the files article.

See appendix C for screenshots of this implementation.

13.3 C01a

This was implemented with a "disclaimer" article in the community name-space of the media wiki, with the following text (in danish):

"The information you read on this page is not formulated by the museum's staff, but the volunteers who have helped us. Although we help the volunteers to provide feedback on each other's contributions, we can never be absolutely sure that the information on the page is completely updated and factually correct."

A link for this page is clearly visible on all artefact articles, to provide a relief to the fear expressed by the curators at the museum in section 5.1.

See appendix D for screenshots of this implementation.

13.4 M07a

A text on the front-page article was formulated, talking advantage of the findings in the "Politiets registerblade" project (described in section 3.1) as well as the findings in the Art Collector project 3.3. A sense of purpose can be established through a narrative telling the members of the crowd, why this is important at how they are making something possible, which would otherwise not be possible.

As mentioned in the requirements section 5.2 the members of crowd has an interest in helping the digitization of coins and medals, to communicate their interest in this type of artefact, for them to recruit new members to their community around the 129 year old association of Danish numismatics.

This has been my focus when writing the following proposal for the text on the frontpage of the Wiki.

Welcome to the development version of wiki.samlinger.natmus.dk
This is one of the National Museum of initiatives to ensure the digitization of everyone's cultural heritage. We need volunteers - maybe you? - To help with the digitization of our extensive archives and collections. It is our hope that we through increased digital representation of our collections can help in promoting digital accessibility to information hiding in the protocols finally hoping to raise awareness of our objects and the many fantastic stories that often hides behind them.
Precisely through storytelling based on the specific subject , we believe in creating a justification for our cultural heritage in the digital society .

The collection of coins and medals
We are currently working to digitize our coins and medals collection, read more about the collection here [a link to an article about the collection].
As a volunteer you can help us by:

1. Writing off protocols with information about coins
2. Create and fill out forms for coins and medals,
3. Share interesting pictures and information, so that together we can spread the word about the project.

Please read our disclaimer and terms of use, before diving into contribution to the project - they are short at simple.

See appendix E for a screenshot of this implementation.

13.5 M10a

I made sure to use a table in the sidebar of the transcription interface, matching the look-and-feel of the MediaWiki.

One way of achieving a consistently looking GUI is to focus on making it simpler, by removing unused features and basically dumbing it down. I used some time investigating the possibility of implementing a custom theme but ended up choosing to change the logo of the Wiki for the logo of the national museum, as well as

As no specific design guidelines apply for the museum, I have chosen not to customize fonts, colours or alike, This requirement might evolve into concrete requests for graphical design changes in the future, in which case it should be re-prioritized before implemented.

All screenshots of the implemented prototype, in the appendices, contain this implementation.

13.6 M10b

The semantic media wiki and semantic forms extensions used for implementation of M02a, has almost no translation into Danish: 277 of 280 and 188 of 195 messages translated respectively. It would be very easy to contribute these translations via

https://translatewiki.net/wiki/Special:Translate/ext-semanticmediawiki?filter=&action=page&language=da

Because of the academic nature of this project, I have chosen not to spend time on translating the extensions – something that I would otherwise have done.

When developing extensions, such as the Transcribe extension implementing **M01a**, any text string has to be moved to a separate *.i18n.php file.

13.7 C12a

National museum's collection's WIKI: About
Here you will find a brief description of the site and a number of conditions and guidelines that we would like all contributors to kept in mind when they contribute content to our collection.

1. **We appreciate your voluntary contributions:** All contributors on this Wiki is from voluntary people, except our curators who sometimes contribute their own knowledge or corrections. Do you feel that at some point we introduce you to tasks that you do not think is fair to ask of a volunteer - please let us know and let us negotiate a mutual beneficial cooperation.
2. **We speak respectfully to each other:** We expect all to have a reasoned and constructive tone then debating articles and all to have an open and welcoming attitude towards newcomers.
3. **The data released into the public domain:** When you contribute content to this Wiki you release it at the same time under the open license, called creative commons zero [a link to https://creativecommons.org/publicdomain/zero/1.0/], meaning anyone can reuse, modify and distribute the content without asking your permission, even for commercial purposes.

In addition, our general disclaimer [an internal link to the disclaimer] holds.

See appendix F for a screenshot of this implementation.

13.8 C08a

As mentioned in section 4.4 about digital asset management systems, it has many benefits to reuse a component for the management of digital media assets across multiple products in an organisations IT-portfolio.

It is possible to interface with the museums DAMS via the CIP web service. This web service has an SDK implemented in PHP which exposes the services an operations to a developer through in-line documentation containing data-type annotations for input parameters and output return values of various methods.

The MediaWiki provides an extensible architecture for handling media files, which has originally been designed to enable the use of media files from remote media-wikies within a particular wiki's articles. Classes central in this extensible architecture are shown in a simplified class-diagram on figure 25, as this is not explicitly showing all attributes and operations of a particular entity, nor does it show all classes of the packages – just the ones relevant for the implementation of this particular feature.

Figure 26 shows a sequence diagram of the MediaWiki utilizing the three implemented classes,

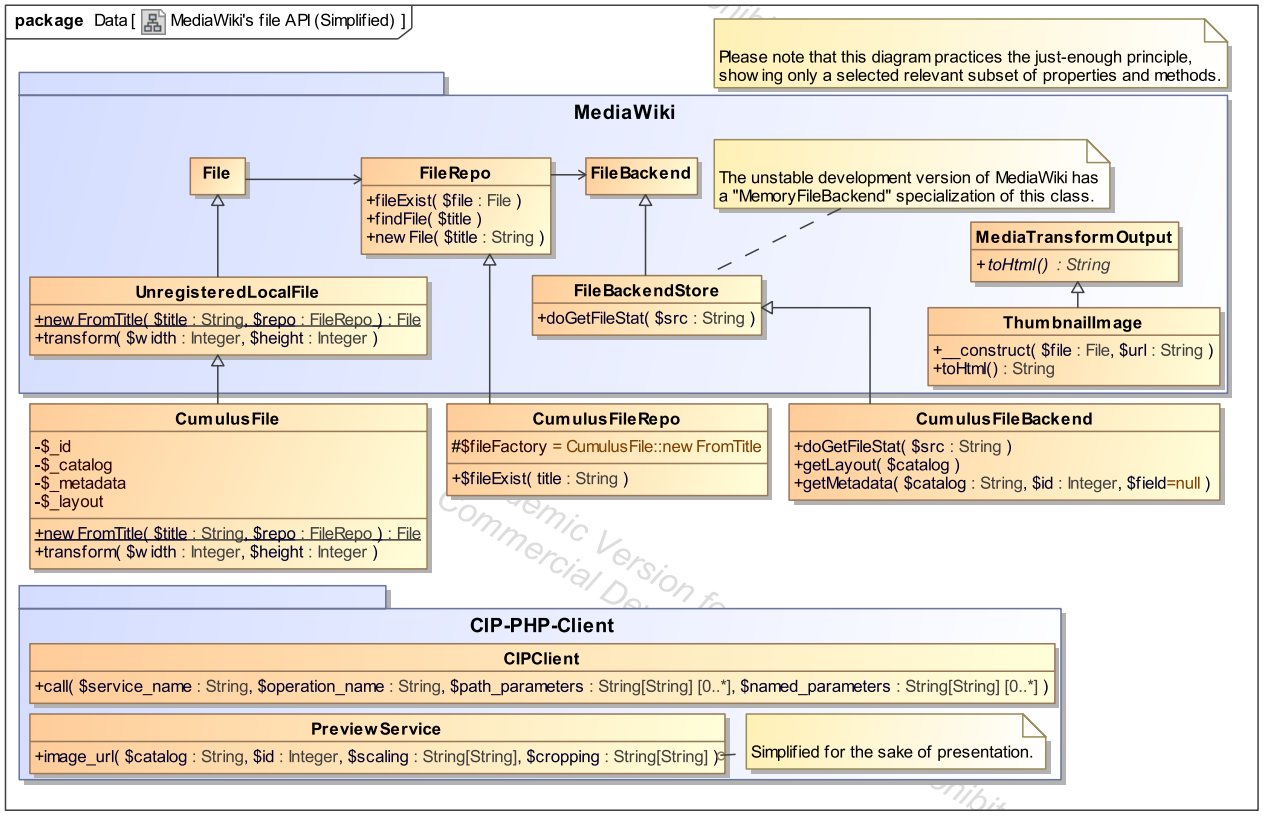


Figure 25: A simplified class-diagram showing the entities and relations of the MediaWiki, the CIP-PHP-Client/SDK and the three classes implemented in this project.

13.9 D06a

This is possible utilising semantic forms and templates. I have set up a description field and a weight field. The actual fields can be changed when people start entering data. It will be evident which fields

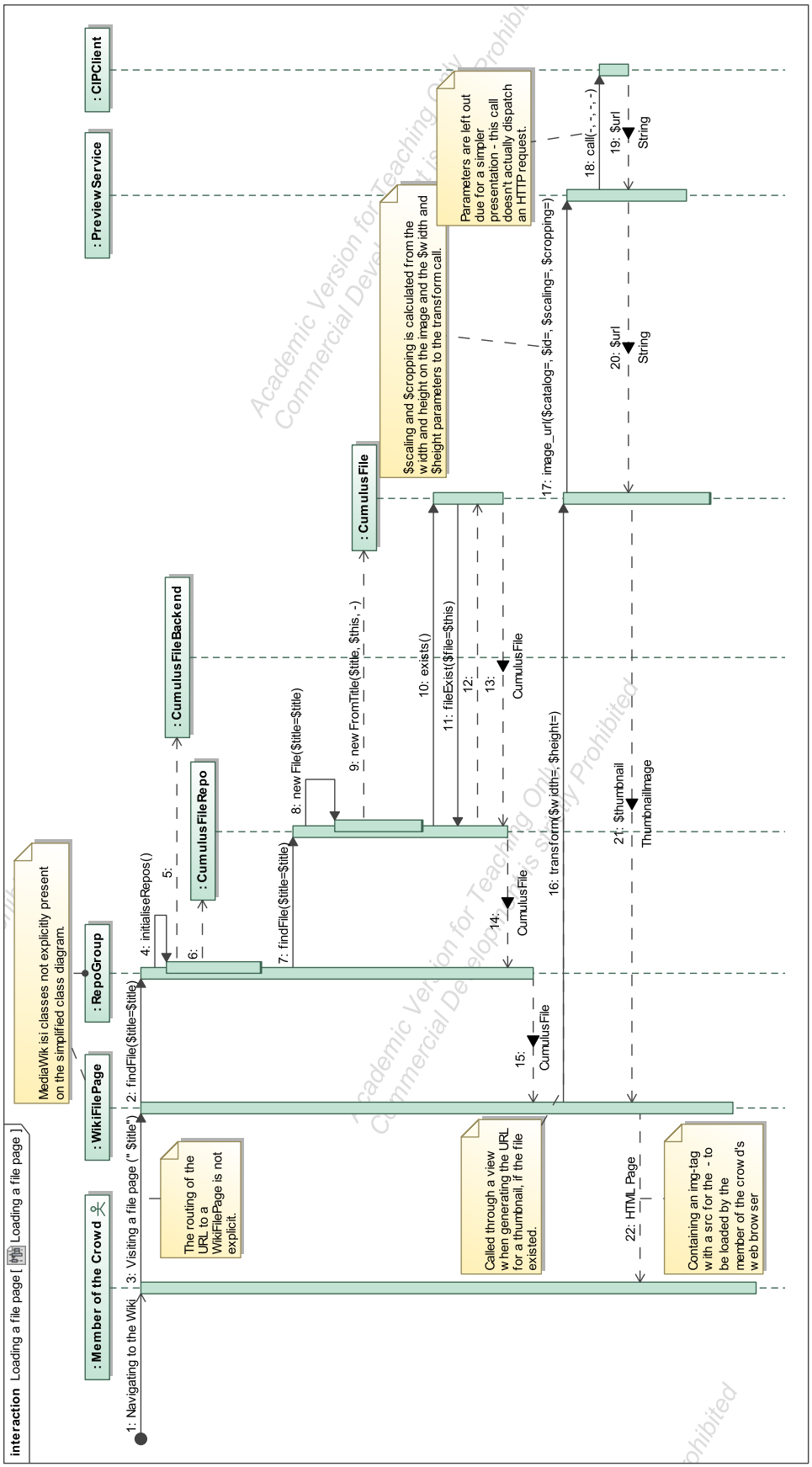


Figure 26: A sequence diagram showing the loading of an asset's file page and the delegation of control throughout the system.

are needed as data that does not fit within the structure will be contributed in the non-structured free text area below the structured fields. I would expect the following fields about coins and medals, from the knowledge of what is written in the protocols and what the numismatics are publishing on their own websites [20]:

- References to various catalogues, such as Hans Henrik Schou, Holger Hede, Frovin Sieg or Georg Galster's [30].
- The demonination - the currency amount visible from the coin.
- Inscriptions on the coin.
- The period / year in which the coin was produced.
- Where the coin was produced.
- Where the coin was found.

[20] One such is Niels Jensen and Mogens Skjoldager's website www.danskmoent.dk

Discussion

The workload estimates presented in the fitness-analysis is based on the result from a planing poker game, where solutions for the various categories of functionality. The solutions presented as arguments for the various estimates could have been discussed in even further depth, as this would bring more credibility to the results of the analysis.

Throughout the implementation of the solution, it became evident that the external services, the CIP installation in particular, was very unstable. Several day long breakdowns and long response-times were a frequent experience. Looking back on the process, it would probably have been a great idea to do an up front assessment of the quality of service of these external components. One such assessment could contain setting up an external system to ping the system on a frequent basis, such as the pingdom.com software-as-a-service solution. The assessment could answer questions about the existing infrastructure. Could they be considered stable for development going forward? Could their configuration be changed to enhance performance? Or could they be substituted for other systems providing similar functionality?

When choosing systems for the fitness analysis, nothing guarantees that the most fit system is in fact considered by the analysis, there might be undiscovered systems that could have been even more fit. The systems presented in the fitness analysis did not really represent systems with similar feature-sets – we could have tried to include even more systems of the same type.

Evaluating the fitness of the candidate systems, this could have been done in respects to each of the user stories, instead of evaluating them against categories of functionality. This would give a more precise estimate, but it would probably make the fitness analysis too costly in terms of time, and as this is an upfront investment providing no notable business value to the projects product owner, this activity should be as minimal as possible. We could also have tried to estimate each of the user stories without a system, to get a benchmark, this would have given the Amazon Mechanical Turk a more representable fitness-score.

I cold have evaluated the fitness directly, instead of estimating workload - maybe on an integer -3 to +3 scala, but I didn't find any proven method of doing this.

I could have considered introduced the components of the infrastructure (such as Cumulus and GenReg) in the analysis, but choose not to as I feared that this would remove focus from the real objective of the analysis, as these components were more or less given from the product owners point of view, due to the strategic need to consolidate infrastructure.

I have implemented the user stories as just-enough, and it is expected that the product still has undiscovered requirements, and as such it is expected that more iterations will be needed for it to fulfil the functional requirements sufficiently. One such just-enough implementation is the transcription interface, that successfully supports the creation of annotations containing transcriptions on an image file, but gets confused when changing existing annotations, as they lead to the duplication of annotations. This is because of the way the API for changing the semantic values of an article, has been implemented.

I could have chosen to focus more on the design and implementation of functionality, risking that I was implementing functionality which would later be changed through the realisation of changed requirements. This is a balance and I found this level of requirements elicitation to design and implementation appropriate for this particular challenge.

When presenting the implementation of user stories, I could have chosen to present a complete class diagram showing the domain model as a result of choosing the MediaWiki, in terms of the MediaWiki's entities and relations, such as articles, files and links. But as this would be presenting a lot of design that I technically did not contribute, I made the choice of leaving it out.

I tried and failed a couple of times on the establishment of an agile process around the creation of this report and the thesis in general, but found this very challenging. I have identified what I think is the three main reasons:

The definition of done, when designing and implementing a software system in a thesis is different - the student has to present a report at the end of the project, which is not typical for a typical agile software project within the industry. The requirement of the hand-in of a report essentially violates the second agile principle of the agile manifesto, namely "Working software over comprehensive documentation." van Bennekum Alistair Cockburn Ward Cunningham Martin Fowler James Grenning Jim Highsmith Andrew Hunt Ron Jeffries Jon Kern Brian Marick Robert C. Martin Steve Mellor Ken Schwaber Jeff Sutherland Dave Thomas [43]

The second reason is the fact that an agile process, such as SCRUM works because it forces members of the development team to engage in conversations with each other and the product owner at events with specifically designed purposes. This is why SCRUM does not deliver when the development team is small.

My third and last insight has to do with the roles of a SCRUM project. Being the product owner of the report, as well as the scrum master trying to establish the process and at the same time being the only member on the development team, introduces conflicts of interest, leading to a less effective process.

Conclusion

Through a reuse-oriented approach to a real-life challenge proposed by the National Museum of Denmark, I have deduced a perception of the museum's context. Trough qualitative interviews and workshops with key stakeholders and representatives of future users of the system, the perception of the challenge, with its context and constraints has been codified in two main artefacts, the section 4 on the existing software platform and a preliminary product backlog, section 7, of user stories presenting an ordered list of requests for functionality in respect to the perceived business-value of the stakeholders at and around the museum.

From the preliminary product backlog I deduced six categories of functionality, with the intention of simplifying an analysis as well as allow for the candidate systems to propose alternative variations on the original request for functionality. This was necessary as software engineering with reused components introduce software which has not originally been designed for the exact challenge at hand. Having functionality in a few abstract categories allows the reused components to inspire the product owner and the developer to change the original requirements.

The MediaWiki, Amazon Mechanical Turk and the Zooniverse's Scribe transcription interface, were analysed by proposing workload estimates on the implementation of each of the six categories of functionality, through the process of planning poker.

The MediaWiki was found as the fittest system with an estimated ~ 70% fitness. Another significant result of the fitness analysis was the discussion of different approaches to implementations of the functionality proposed by the preliminary product backlog. A modification of requirements was effectuated from the new knowledge introduced by the selection of the MediaWiki as foundation of the solution.

Finally the design and implementation of solutions to user stories were proposed and implemented, one user story at a time, in a single agile sprint. Most significantly a generic and reusable user interface for transcription of image files on a MediaWiki and the integration of the MediaWiki with the museum's digital asset management system Cumulus, through the Canto Integration Platform was implemented.

I have thoroughly examined the requirements to a digital platform for crowd engagement, in the museum's context and concludes that the MediaWiki fits these requirements to a large degree, but is not

an absolute fit system. I have proposed modification to the system through modified requirements and then executed the design and implementation of the nine most valuable user stories, showing how the MediaWiki is integrated with the existing infrastructure of the museum, in particular its digital asset management system.

The solution has not been tested on users, thus the actual fitness of the system to the concrete challenge, has not been formally verified. The components integrated to solve the challenge are successfully solving similar challenges in comparable contexts, such as the Zooniverse and WikiPedia projects. This is why I conclude that is safe to assume that the designed and implemented solution supports the essential activities required of a platform for crowd engagement in the context of the National Museum of Denmark.

Bibliography and appendices

References

[1] Inc. Amazon Web Services. *Amazon Mechanical Turk - Sample Code & Libraries*, 2014.
URL: `https://aws.amazon.com/code/Amazon-Mechanical-Turk/`.

[2] Jeppe Christensen. *BIBTEKKONF*, October 2013.
URL: `http://prezi.com/6jpse9bjonue/bibtekkonf/`.

[3] Mike Cohn. *Non-functional Requirements as User Stories*, November 2008.
URL: `http://www.mountaingoatsoftware.com/blog/non-functional-requirements-as-user-stories/`.

[4] Mike Cohn. *Advantages of the "As a user, I want" user story template.*, April 2008.
URL: `http://goo.gl/LBGvHI`.

[5] The MediaWiki Community. *Extension:Data Transfer - MediaWiki*, 2014.
URL: `https://www.mediawiki.org/wiki/Extension:Data_Transfer`.

[6] The MediaWiki Community. *Extension:External Data - MediaWiki*, 2014.
URL: `https://www.mediawiki.org/wiki/Extension:External_Data`.

[7] The MediaWiki Community. *Extension:FlaggedRevs - MediaWiki*, 2014.
URL: `https://www.mediawiki.org/wiki/Extension:FlaggedRevs`.

[8] The MediaWiki Community. *Extension:MultiBoilerplate - MediaWiki*, 2014.
URL: `https://www.mediawiki.org/wiki/Extension:MultiBoilerplate`.

[9] The MediaWiki Community. *Extension:PreloadManager - MediaWiki*, 2014.
URL: `https://www.mediawiki.org/wiki/Extension:PreloadManager`.

[10] The MediaWiki Community. *Extension:Proofread Page - MediaWiki*, 2014.
URL: `https://www.mediawiki.org/wiki/Extension:Proofread_Page`.

[11] The MediaWiki Community. *Extension:Semantic Forms - MediaWiki*, 2014.
URL: `https://www.mediawiki.org/wiki/Extension:Semantic_Forms`.

[12] The MediaWiki Community. *Extension:Semantic MediaWiki - MediaWiki*, 2014.
URL: `https://www.mediawiki.org/wiki/Extension:Semantic_MediaWiki`.

[13] The MediaWiki Community. *Extension:SocialProfile - MediaWiki*, 2014.
URL: `https://www.mediawiki.org/wiki/Extension:SocialProfile`.

[14] The MediaWiki Community. *Manual:Creating pages with preloaded text - MediaWiki*, 2014.
URL: https://www.mediawiki.org/wiki/Manual:Creating_pages_with_preloaded_text.

[15] The MediaWiki Community. *Doxygen generated documentation for MediaWiki*, 2014.
URL: https://doc.wikimedia.org/mediawiki-core/master/php/html/.

[16] The MediaWiki Community. *Extension Matrix - MediaWiki*, 2014.
URL: https://www.mediawiki.org/wiki/Extension_Matrix.

[17] The MediaWiki Community. *Category:Extensions by category - MediaWiki*, April 2014.
URL: https://www.mediawiki.org/wiki/Category:Extensions_by_category.

[18] The Semantic MediaWiki Community. *Help:Semantic templates - semantic-mediawiki.org*, 2014.
URL: https://www.semantic-mediawiki.org/wiki/Help:Semantic_templates.

[19] The Semantic MediaWiki Community. *Help:Inline queries - semantic-mediawiki.org*, 2014.
URL: https://semantic-mediawiki.org/wiki/Ask.

[20] The Semantic MediaWiki Community. *Semantic Forms - semantic-mediawiki.org*, April 2014.
URL: https://www.semantic-mediawiki.org/wiki/Semantic_Forms.

[21] The Semantic MediaWiki Community. *Help:Using SPARQL and RDF stores - semantic-mediawiki.org*, 2014.
URL: https://semantic-mediawiki.org/wiki/Help:Using_SPARQL_and_RDF_stores.

[22] The WikiPedia Community. *Wikipedia:Spellchecking - Wikipedia, the free encyclopedia*, 2014.
URL: https://en.wikipedia.org/wiki/Wikipedia:Spellchecking.

[23] Pete Deemer and Gabrielle Benefield. *The Scrum Primer v.2.0. A Lightweight Guide to the Theory and Practice of Scrum.* 2012.
URL: http://www.scrumprimer.org/scrumprimer20.pdf.

[24] Roy T. Fielding. *Software Architectural Styles for Network-Based Applications.* 2009.

[25] Andrea Forte and Amy Bruckman. *Why do people write for Wikipedia? Incentives to contribute to open–content publishing. Proc. of GROUP*, 5:6–9, 2005.
URL: http://jellis.org/work/group2005/papers/forteBruckmanIncentivesGroup.pdf.

[26] Justin Gillis. *Retrieving the Weather of the Past*, October 2012.
URL: http://green.blogs.nytimes.com/2012/10/24/retrieving-the-weather-of-the-past/.

[27] Jan Friso Groote and Mark van den Brand. *Software engineering: Redundancy is key. Science of Computer Programming*, pages –, 2013. ISSN 01676423, 18727964. doi: 10.1016/j.scico.2013.11.020.

[28] Google Inc. *Google Trends - Web Search interest: crowdsourcing - Worldwide, 2004 - present*, 2014.
URL: http://www.google.dk/trends/explore#q=crowdsourcing.

[29] Charlotte S. H. Jensen. *Digitalisering og crowdsourcing af møntsamlingen*, March 2013.
URL: http://digital.natmus.dk/2013/05/29/digitalisering-i-montsamlingen/.

[30] Niels Jørgen Jensen and Mogens Skjoldager. *Kataloger over danske mønter*, May 2014.
URL: http://www.danskmoent.dk/katalog.htm.

[31] Kulturstyrelsen. *Kulturministeriets digitaliseringsstrategi*, Januar 2012.
URL: http://www.kulturstyrelsen.dk/fileadmin/publikationer/andre_publikationer/Kulturministeriets_digitaliseringsstrategi_2012-2015._farver.pdf.

[32] Ellen Lee. *As Wikipedia moves to S.F., founder discusses planned changes*, November 2007.
URL: http://www.sfgate.com/business/article/As-Wikipedia-moves-to-S-F-founder-discusses-3233536.php.

[33] National Museum of Denmark. *Mission og vision - Nationalmuseet*, 2014.
URL: http://natmus.dk/nationalmuseet-som-organisation/opgaver-maal-og-historie/mission-og-vision/.

[34] The Royal Library of Denmark. *DANMARK SET FRA LUFTEN - Før Google*, September 2011.
URL: http://www.kb.dk/danmarksetfraluften/.

[35] Arfon of The Zooniverse Blog. *Making the Zooniverse Open Source*, February 2013.
URL: http://blog.zooniverse.org/2013/02/18/making-the-zooniverse-open-source/.

[36] Dimitris Paraschakis. *Crowdsourcing cultural heritage metadata through social media gaming*. Master's thesis, Malmö University, Department of Computer Science., 2013.

[37] Daniel. Schall. *Service-Oriented Crowdsourcing : Architecture, Protocols and Algorithms*. Springer New York, 2012. ISBN 1461459567, 1461459559, 9781461459569, 9781461459552.

[38] Ian Sommerville. *Software Engineering 9th Edition*. Pearson, 2011. ISBN 0137053460, 9780137053469.

[39] Camilla Stockmann. *Nationalmuseet lukker udstilling af hemmelige årsager*, March 2014.
URL: http://politiken.dk/kultur/ECE2224102/nationalmuseet-lukker-udstilling-af-hemmelige-aarsage

[40] Canto Systems. *Cumulus Query Format*, 2014.
URL: http://samlinger.natmus.dk/CIP/doc/QueryFormat.html.

[41] Canto Systems. *Cumulus Query Format*, 2014.
URL: http://samlinger.natmus.dk/CIP/doc/CIP.html.

[42] The. *Manual:Hooks/EditPage::showEditForm:initial - MediaWiki*, 2014.
URL: https://www.mediawiki.org/wiki/Manual:Hooks/EditPage::showEditForm:initial.

[43] Kent Beck Mike Beedle Arie van Bennekum Alistair Cockburn Ward Cunningham Martin Fowler James Grenning Jim Highsmith Andrew Hunt Ron Jeffries Jon Kern Brian Marick Robert C. Martin Steve Mellor Ken Schwaber Jeff Sutherland Dave Thomas. *Manifesto for Agile Software Development*.
URL: http://agilemanifesto.org/.

[44] Luis von Ahn and Laura Dabbish. *Designing games with a purpose*. *COMMUNICATIONS OF THE ACM*, 51(8):58–67, 2008. ISSN 00010782, 15577317. doi: 10.1145/1378704.1378719.

[45] Bill Wake. *Independent Stories in the INVEST Model*, February 2012.
URL: http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/.

[46] Jacob Riddersholm Wang. *Det Digitale Nationalmuseeum Strategi (Offentlig) (2012-2015)*, September 2012.
URL: http://digital.natmus.dk/strategi/.

[47] Don Wells. *User Stories*, 1999.
URL: http://www.extremeprogramming.org/rules/userstories.html.

[48] The Wi. *List of crowdsourcing projects*, 2014.
URL: https://en.wikipedia.org/wiki/List_of_crowdsourcing_projects.

[49] Michał Wojciechowski. *imgAreaSelect - image selection/cropping jQuery plugin - odyniec.net*, 2014.
URL: http://odyniec.net/projects/imgareaselect/.

[50] Zooniverse. *Zooniverse on GitHub*, 2014.
URL: https://github.com/zooniverse.

List of Figures

List of appendices

A Source code for the prototype

For the source code to the extensions, please visit

http://kraenhansen.dk/masters-thesis/

B Screenshots of the prototype - implementing M02a

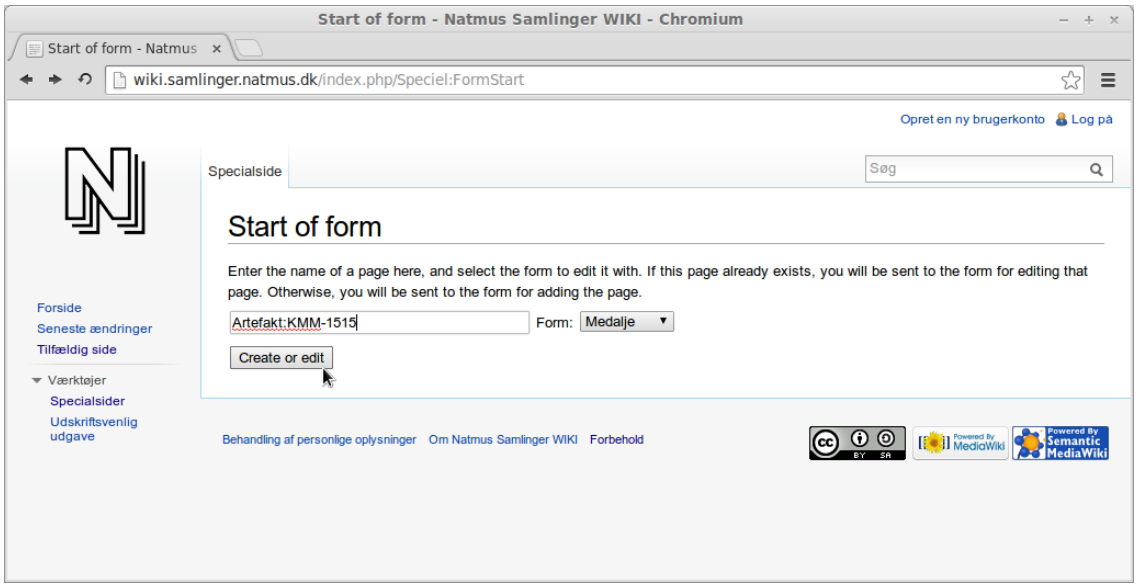


Figure 27: Screenshot showing the “Start of form” interface used when creating an article with a form, the user enters a name and selects the correct form.

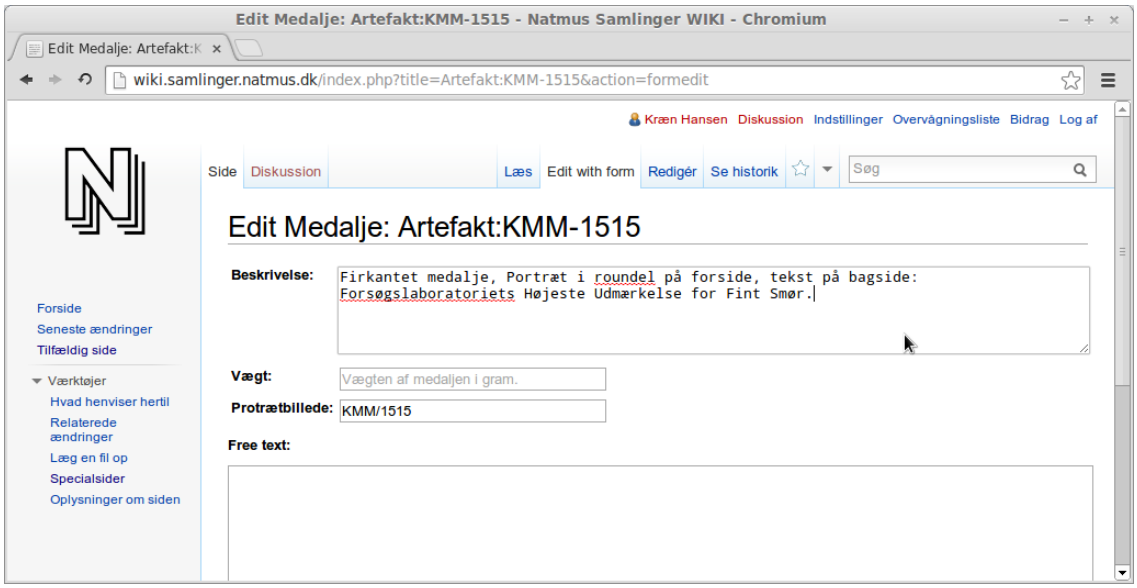


Figure 28: The user enters the appropriate values, if known at this moment and saves the form.

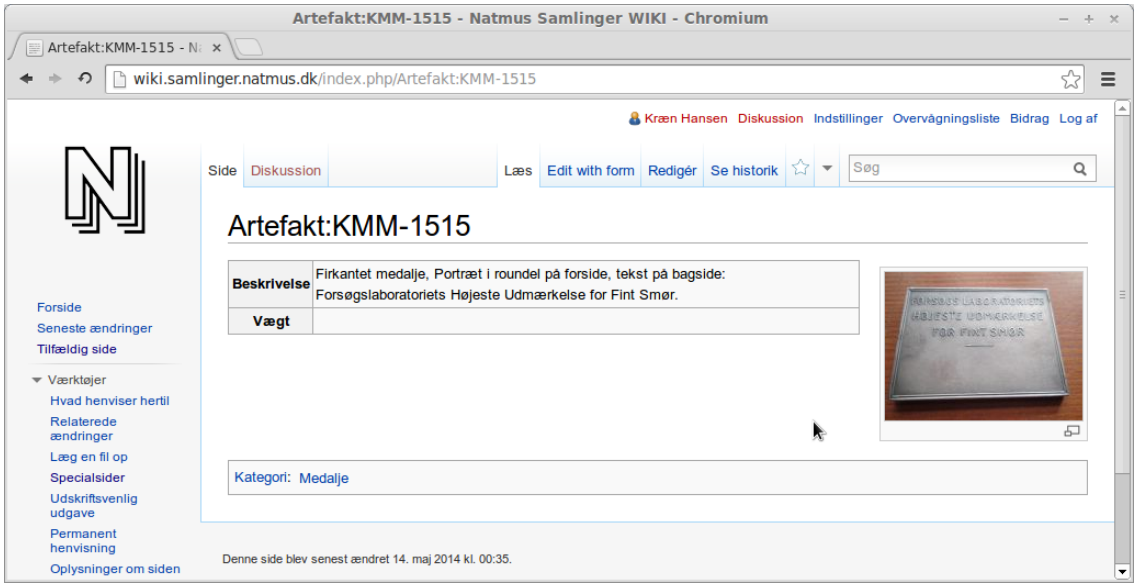


Figure 29: The article for the artefact is created and shown to the user.

C Screenshots of the prototype - implementing M01a

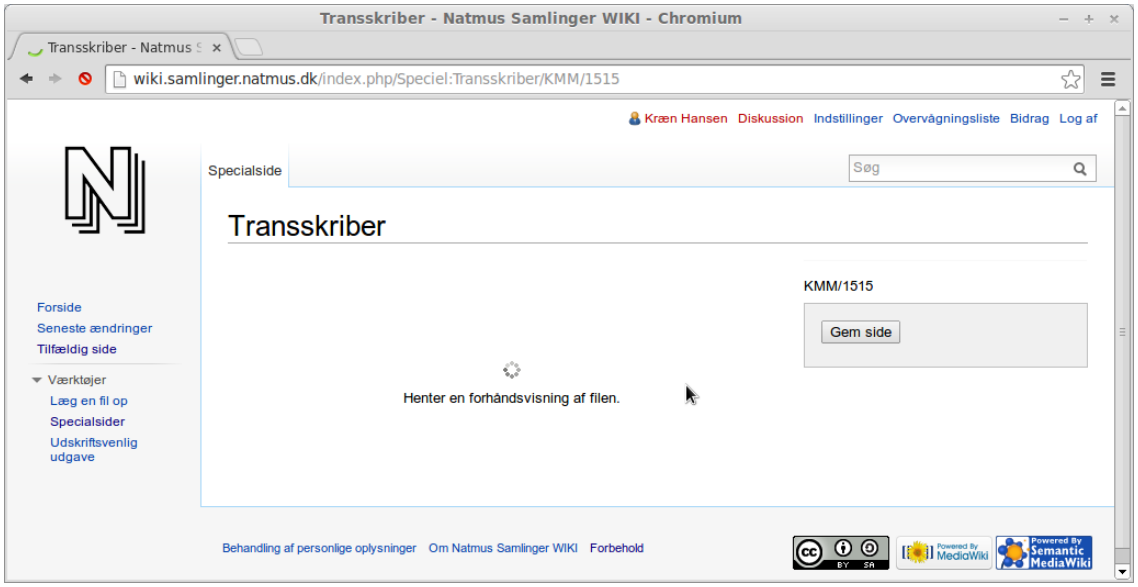


Figure 30: Screenshot showing loading of the image preview on the transcription interface.

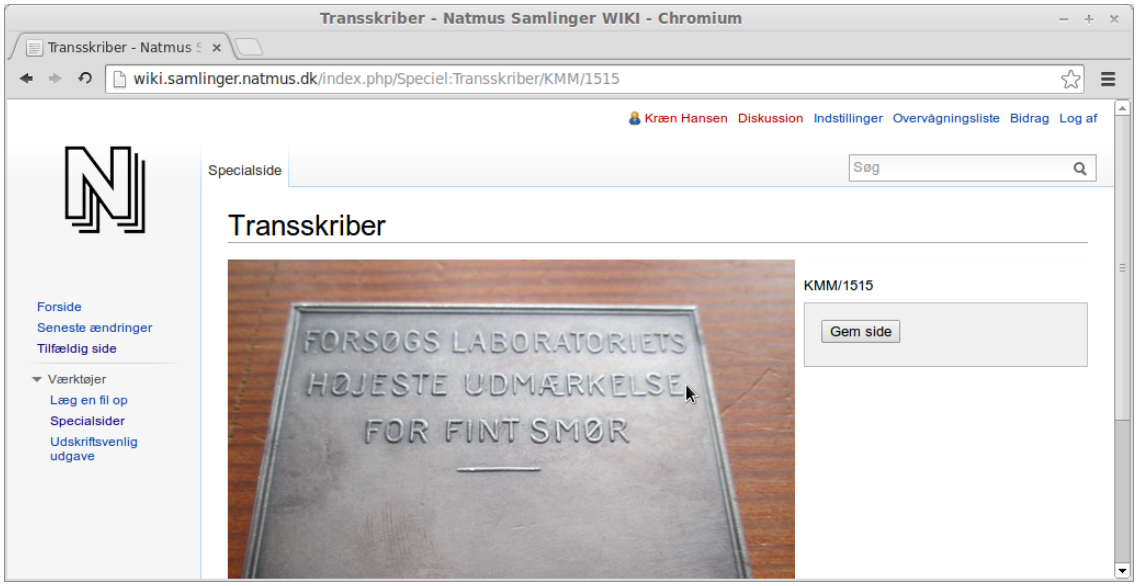


Figure 31: Screenshot showing the transcription interface on an asset which has not been transcribed.

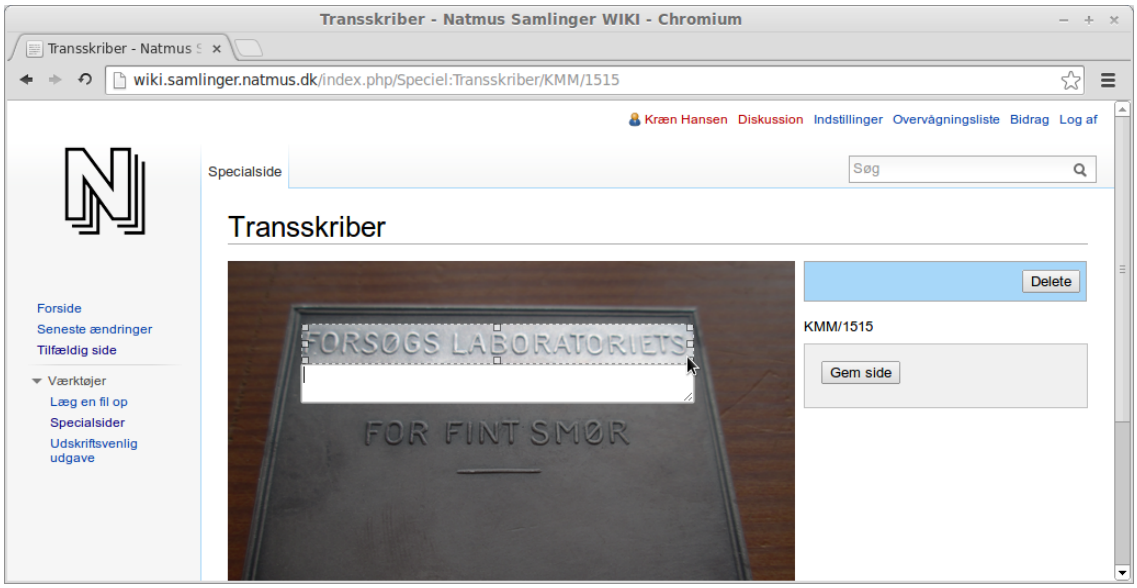


Figure 32: Drag-dropping on the image makes a selection with a textarea below, in which the user types the content of the transcription.

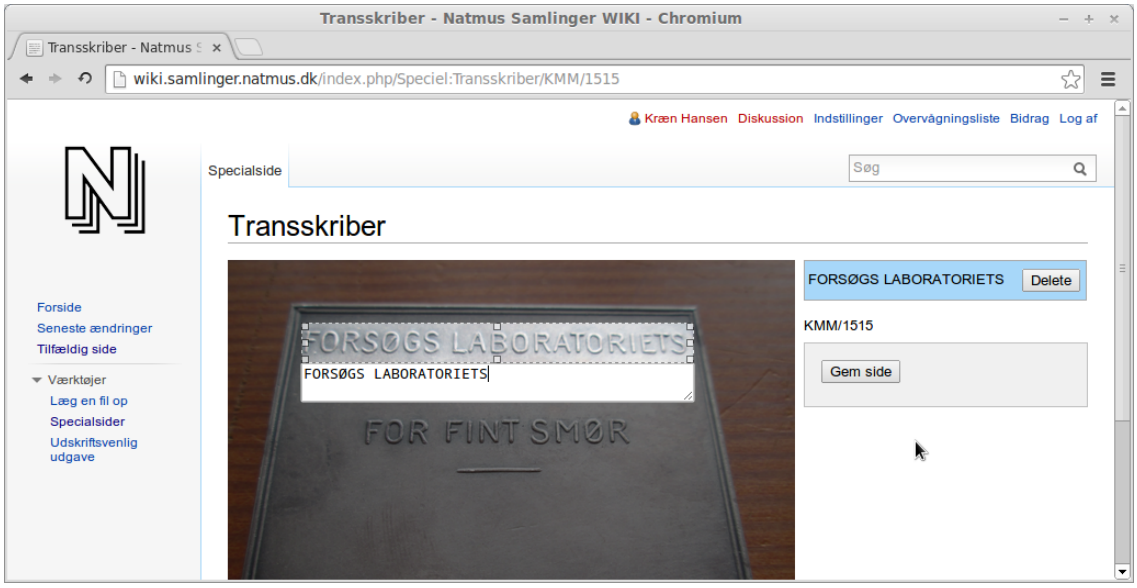


Figure 33: When the user types in the text area the table to the right of the image is updated accordingly.

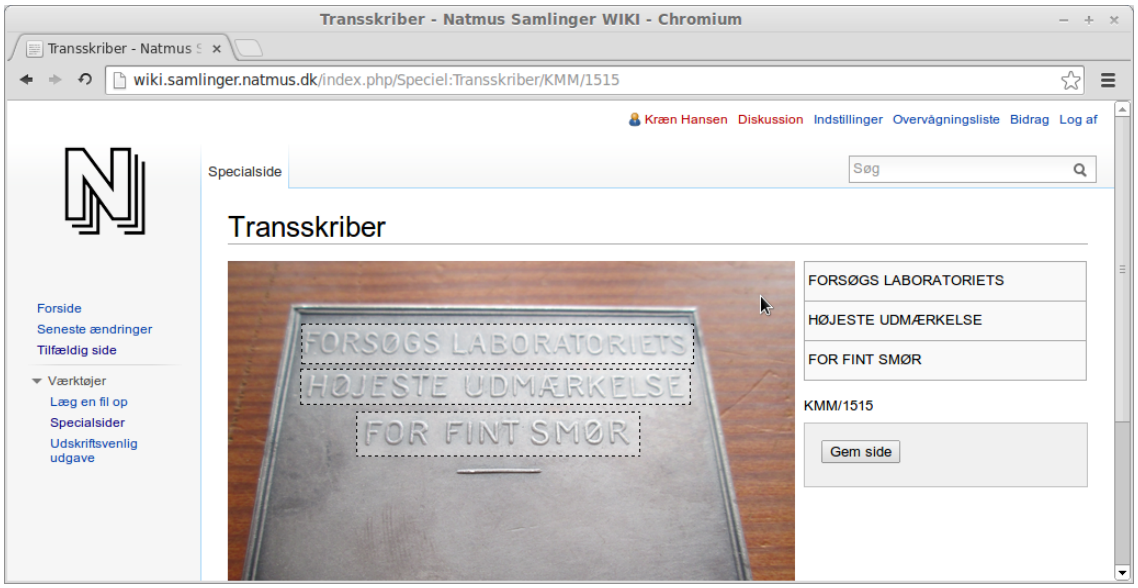


Figure 34: An image showing the interface when the selection has been deselected and the preview image is hovered by the cursor.

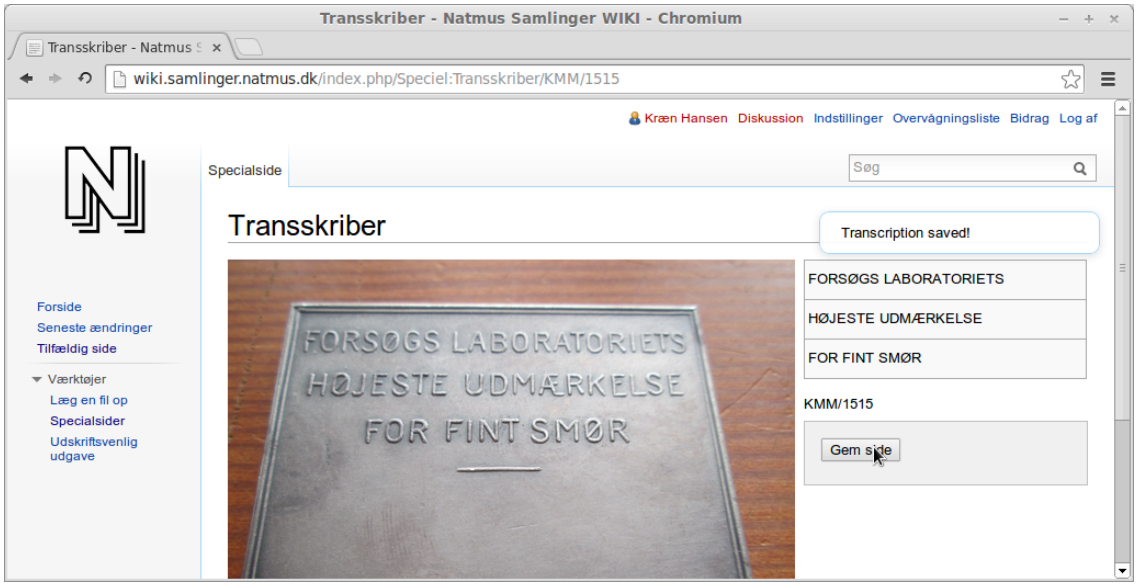


Figure 35: When hitting the save button, a notification confirms that the transcription has been saved.

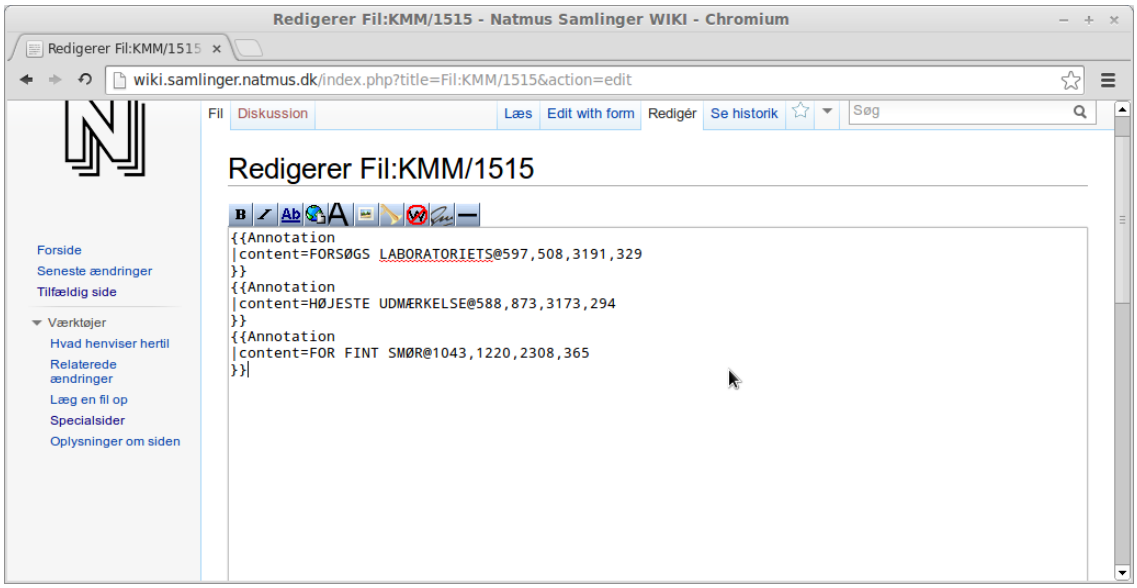


Figure 36: Screenshot showing the transcribed files article in edit mode, with its new semantic properties (created through the transcription GUI).

D Screenshots of the prototype - implementing C01a

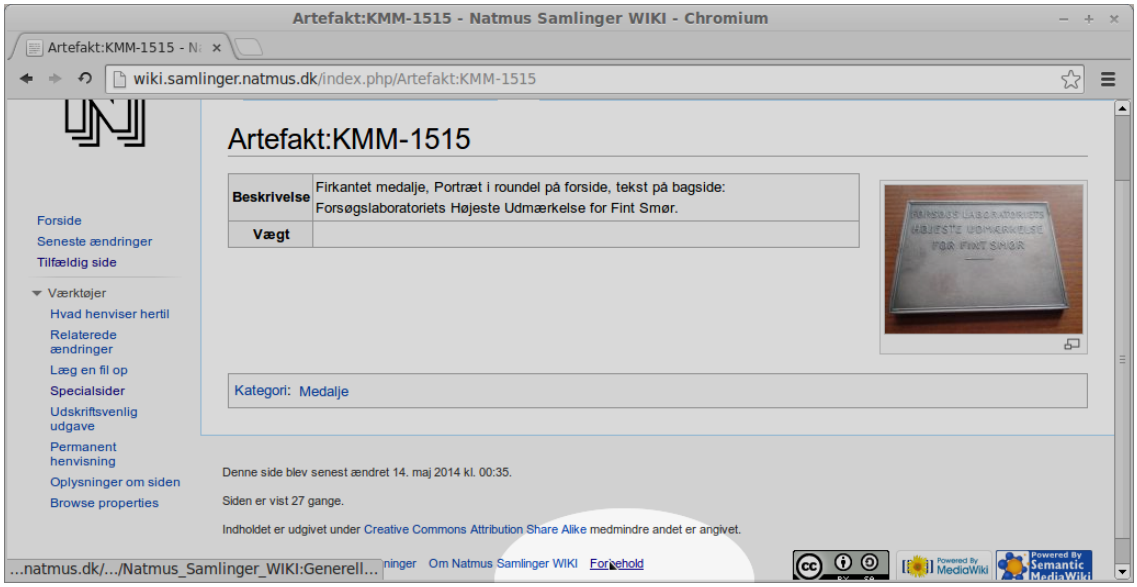


Figure 37: The disclaimer link is shown in any artefact.

Figure 38: The disclaimer showing solving user story C01a.

E Screenshots of the prototype - implementing M07a

Figure 39: The frontpage with an encouraging message to the members of crowd.

F Screenshots of the prototype - implementing C12a

Figure 40: The about page with guidelines to the members of crowd.