Designing Semantic Enrichment for non-expert users in a News Outlet website

Master's thesis

Magdalini Fotiadou

August 2016
Abstract

Semantic Enrichment, a lower-case approach to the Semantic Web, can be defined as adding contextual information to existing content in order to enable classification, disambiguation, indexing and data repurposing. Direct benefits of semantically enriching data are increased visibility, content attribution, and tailored presentation, both when it comes to SERPs (search engine result pages), and social media. Moreover, uniquely identifying, structuring and linking data may give rise to various applications such as personalized recommendations and semantic search.

This Master’s thesis explores how Semantic Enrichment can be performed by non-expert users in the context of Titan.uio.no, the news outlet of the faculty of Mathematics and Natural Sciences in the University of Oslo. Since the website is powered by Drupal 7, Metatag, an existing Drupal module has been assessed and customized, following a user-centered design paradigm.
Acknowledgements

Upon the completion of this master thesis I would like to thank Gisle Hannemyr for his supervision, thorough explanations, and continuous availability.

I am indebted to the editorial team, the authors, and the bloggers behind Titan.uio.no who patiently tested with me and showed interest in the subject. Thank you for offering your time and your invaluable feedback.

Finally I would like to express my deep gratitude to all those who supported me this year: my family for helping me in every way conceivable, my friends and flatmates for their persistent encouragement and curiosity, and Knut, for proofreading the translation of the interface and listening to my incessant monologues.
# Table of Contents

**Abstract** ........................................................................................................ III

**Acknowledgements** ...................................................................................... V

**Table of Contents** .......................................................................................... VII

**Table of Figures** ............................................................................................ XI

**Tables** ........................................................................................................... XI

1. Introduction ........................................................................................................ 1

1.1 Motivation ........................................................................................................ 1

1.1.1 The Semantic Web vision ........................................................................ 1

1.1.2 Semantic Enrichment ............................................................................... 2

1.2 Problem Area ................................................................................................ 2

1.2.1 Titan.uio.no ............................................................................................ 2

1.2.2 Drupal ...................................................................................................... 3

1.2.3 User roles and challenges ......................................................................... 3

1.2.3.1 Necessity of the module .................................................................... 5

1.3 Research Questions ....................................................................................... 5

1.4 Activity Agenda ............................................................................................ 5

1.5 Outline ........................................................................................................... 5

2. Background ...................................................................................................... 7

2.1 The Semantic Web Stack .............................................................................. 7

2.1.1 RDF ......................................................................................................... 8

2.1.2 URIs ........................................................................................................ 8

2.1.3 Ontologies ................................................................................................ 9

2.1.3.1 Ontology Languages and frameworks ............................................. 9

2.1.3.2 Vocabularies .................................................................................... 10

2.1.4 Semantic Annotation Formats .................................................................. 17

2.1.4.1 RDFa ............................................................................................... 18

2.1.4.2 Microdata ........................................................................................ 19

2.1.4.3 Microformats ................................................................................... 20

2.1.4.4 JSON-LD ........................................................................................ 21

2.1.4.5 Adoption ......................................................................................... 22

2.1.4.6 Upper versus Lowercase Semantic Web ........................................... 23

2.1.4.7 Metadata Standards ......................................................................... 23

2.1.5 Publishing tools and storage approaches ................................................. 24

2.1.5.1 Triplestores, SPARQL, RDB2RDF .................................................. 24

2.1.5.2 APIs, Textual data ............................................................................ 25

2.1.6 Logic, Rules, Inference ............................................................................ 26

2.2 Drupal and the Semantic Web ....................................................................... 27

2.2.1 Core Modules ........................................................................................ 27

2.2.2 Contributed Modules .............................................................................. 28

2.2.2.1 Meta tags ......................................................................................... 29

2.2.3 Desired features for a module used by inexperienced users ................... 31

2.2.4 The future: Drupal 8 ................................................................................ 32

2.3 Conclusion ................................................................................................... 32
5.4.2.3 Iteratively changing titles, labels, descriptions, default values, adding widgets, translating ............................................................. 59
5.4.2.4 Adding validation functions .................................................. 60
5.4.2.5 jQuery: Prepopulating, Collapsing, Vertical Tabs .................... 61
5.4.2.6 Gamification elements .......................................................... 64
5.4.2.7 Advanced permissions .......................................................... 65
5.4.2.8 Content types available for meta-tagging ............................... 66
5.4.2.9 Help material ..................................................................... 67
5.5 Output ...................................................................................... 67
5.6 Conclusion ............................................................................... 68
6. FINDINGS .................................................................................. 70
6.1 Open-Ended, Semi-Structured Interviews .................................. 70
   6.1.1 Sample, Interview Guide and Context .................................... 70
   6.1.2 Thematic Analysis ............................................................... 71
   6.1.3 Implications ....................................................................... 73
6.2 Thinking Aloud ......................................................................... 74
   6.2.1 Sample, Context, Task List .................................................. 74
   6.2.2 Thematic Analysis ................................................................ 75
   6.2.3 Individual Reflections ............................................................ 78
      6.2.3.1 Thinking aloud with Project Leader ................................. 78
      6.2.3.2 Thinking aloud with Communication responsible ............. 79
      6.2.3.3 Thinking aloud with very experienced Author ................. 79
      6.2.3.4 Thinking aloud with inexperienced Blogger ..................... 80
      6.2.3.5 Thinking aloud with experienced Author ......................... 80
   6.2.4 Iterative Approach and Implications for Design .................... 81
   6.2.5 Anticipations and Results ..................................................... 82
6.3 User Acceptance testing .......................................................... 82
6.4 Conclusion ............................................................................... 83
7. DISCUSSION ............................................................................. 84
   7.1 Answering the Research Questions ......................................... 84
      7.1.1 What are the requirements for developing an understandable module that will enable Semantic Enrichment for users with no expertise in the field of the Semantic Web? .................................................. 84
      7.1.2 What are the technical challenges faced? .............................. 85
      7.1.3 What impedes the adoption of Semantic Enrichment by non-experts? Why is the Semantic Web underused? ................................. 86
7.2 The perceived and theoretical benefits of Semantic enrichment in this use case ............................................................. 88
7.3 Future steps ............................................................................. 88
8. CONCLUSION ........................................................................... 90
BIBLIOGRAPHY ............................................................................ 93
APPENDIX A: OPEN-ENDED, SEMI-STRUCTURED INTERVIEWS .......... 100
Interview with Product Manager (Gisle Hannemyr) .......................... 100
Interview with Project Leader (Gunhild Merete Haugnes) ................. 108
Interview with Communication Responsible (Ida Marie Bjørknes) ....... 113
APPENDIX B: THINKING ALOUD SESSIONS .................................. 119
Table of Figures

Figure 1: The Semantic Web stack ................................................................. 7
Figure 2 Publishing and Storage ................................................................. 26
Figure 3 Methods, Artifacts and Analysis ............................................... 42
Figure 4 Workflow ...................................................................................... 53
Figure 5 Token Browser ............................................................................ 61
Figure 6 Unlocking a set of meta tags ......................................................... 65
Figure 7 Enabling the advanced permission setting ................................... 66
Figure 8 Specifying the advanced permissions .......................................... 66
Figure 9 Selecting content types to enrich ................................................ 67
Figure 10 The module as it appears in the publication options area, seen from the admin interface ............................................................ 119
Figure 11 The submodules - meta tag categories ....................................... 120
Figure 12 The vertical tab element .............................................................. 123
Figure 13 The collapsed module as it appears to an editor, before Gamification elements are added ......................................................... 137
Figure 14 Gamification status message ...................................................... 137
Figure 15 Collapsed module interface after Gamification conditions apply 138
Figure 16 Basic Meta tags for Search Engines ........................................... 138
Figure 17 Open Graph (Facebook) Meta tags I ........................................... 139
Figure 18 Open Graph (Facebook) Meta tags II ........................................ 140
Figure 19 Facebook (Open Graph) Page type ............................................ 141
Figure 20 Twitter Cards (Twitter) Meta tags I .......................................... 142
Figure 21 Twitter Cards (Twitter) Meta tags II .......................................... 143
Figure 22 Twitter Cards: Card Types ........................................................... 143
Figure 23 Dublin Core (International Standards) I .................................... 144
Figure 24 Dublin Core (International Standards) II .................................. 145
Figure 25 Google Plus (Google) I ............................................................... 146
Figure 26 Google Plus (Google) II .............................................................. 147
Figure 27 Google Plus (Google): Item types ............................................. 147
Figure 28 Google News ............................................................................ 148

Tables

Table 1 Dublin Core Elements .................................................................. 14
Table 2 Sub-modules, fields, functions ..................................................... 58
Table 3 Prepopulating the meta tag fields ................................................. 63
Table 4 Open-ended, semi-structured interviews: Thematic table for the resulting from Open Coding ................................................................. 71
Table 5 Thematic table from thinking aloud sessions ................................. 76
1. Introduction

This chapter defines Semantic Enrichment, identifies the problem area and the target user group, and states the reasons that necessitate this work. The research questions and activity agenda that will guide the trajectory of the thesis are also outlined here.

1.1 Motivation

1.1.1 The Semantic Web vision

The Semantic Web as envisioned by Berners-Lee, Hendler, and Lassila (2001) describes the expansion of a web consisting largely of documents intended for human consumption, to one that includes data and information that can be manipulated and understood by computers. In their influential article, they (ibid.) provide a series of scenarios that illustrate how agents understand and act upon a vast collection of formalized knowledge. In the Semantic Web, interoperability, intelligent search and combination are achieved as ontologies impose graph-based structures, and apply logic on top of unambiguous pointers to web resources. Formal semantics clearly specify which conclusions can be drawn from the collected information. Linked data is a prerequisite for large scale integration and reasoning on the web (W3C, 2015). This approach is thus different to the “Web 2.0” notion, which revolves around a social graph that includes arbitrary and ambiguous human-generated networks and vocabularies.

Since large scale agent based meditation has not yet occurred Shadbolt, Hall, and Berners-Lee (2006) argued that the Semantic Web has failed to deliver. This assertion is extended to the fact that even though the fundamental technologies have reached maturity and are ready to be deployed outside academia and into production, the ordinary user has not taken advantage of their full potential (Konstantinou, Spanos, Stavrou, & Mitrou, 2010). It has also been claimed that the initial vision of a ubiquitous Semantic Web requires further advances in Artificial Intelligence and Natural Language Processing. According to Hitzler, Krotzsch, and Rudolph (2011) these ambitious expectations “repelled some communities”. A realistic redefinition would hold the Semantic Web as an evolution that “enables machines to access more information that hitherto required human time and attention” (Hitzler et al., 2011).
1.1.2 Semantic Enrichment

Semantic Enrichment implies augmenting the content and context of data by tagging, categorizing, and classifying data in relation to each other, to dictionaries, and/or other base reference sources. At its simplest, this means adding additional contextual information to some existing data set (IBM). Seen from this perspective, Semantic Enrichment is in accordance with the lowercase semantic web technologies that simply aim at encoding data that will be accessed by existing technologies and created by unsophisticated users (Di Iorio, Musetti, Peroni, & Vitali, 2012). The main benefit will be increased visibility (Kidd, 2007) and content attribution. It may also be part of a SEO and social media strategy for a website (Williams, 2012). A stricter approach compliant with uppercase practices (where formal specifications like OWL and RDF are used to represent ontologies and instances of metadata see p. 22) would add the benefits of disambiguation, classification, indexing, accurate data integration, reuse and repurposing, and semantic search that focuses on the meaning behind the query and content.

Since this thesis aims to enable ordinary non-end user to perform Semantic Enrichment, the former “lowercase” approach becomes the focal point, and emphasis is placed on annotating the content with a set of common meta tags while offering as simple a user interface as possible. The motivation for selecting the website described in the following section was testing the research questions in a production website with actual users, in order to get empirical findings.

1.2 Problem Area

1.2.1 Titan.uio.no

Titan.uio.no is the news outlet for science and technology at the University of Oslo, and is affiliated with the faculty of Mathematics and Natural Sciences. The editorial team provides content on a daily basis on disciplines like natural sciences, space science, energy and environment, health, technology, innovation and education. Bloggers also contribute content, while RSS feeds from external news outlets (tu.no, krono, gemini, forskning.no, other blogs) are included. The website aims to draw attention to the faculty and establish a social media presence as it is connected with a Facebook and Twitter account.
1.2.2 Drupal

Titan.uio.no operates on Drupal 7. Drupal is a free and open source content management framework that powers more than one million websites\(^\text{1}\).

Content Management Systems do not only manage textual and multimedia content, but also carry information about the structure and content model of a website. Exposing this to the corpus of semantic information would require considerable expertise. Current CMSs run on traditional Web application servers, usually backed by relational databases (NoSQL approaches do exist). Approaches that do not aim to replace well established infrastructures, but build on them with minimal intrusion and maximal reuse, lower the barrier for entering the Semantic Web for online communities (Corlosquet, Delbru, Clark, Polleres, & Decker, 2009).

Drupal is often singled out due to its flexibility, modularity, extensibility and wide adoption. Since, it “hides the complexity of elements of the Semantic Web from the end user” (Das et al., 2009) it is more appropriate that the common approach of a Java-based site driven by a triplestore. Drupal natively supports RDFa and has included in its core the RDF module. At the same time, a multitude of contributed modules add semantic functionality. Drupal 8 supports RDFa markup and uses the popular Schema.org vocabulary, but there are no plans for upgrading Titan.

1.2.3 User roles and challenges

A fundamental feature of Drupal is refined access control. The administrator may set permissions for different types of users in order to define who can do what for core features and contributed modules. Out of the box, Drupal recognizes the three following types of users:

1. User/1, also known as the super user, is the account one is prompted to create immediately after successful installation. This account is unique to the website and different from all other users in the installation as it has no permissions limitations.
2. User/0 represents anonymous or unregistered users.
3. User/2 and all other registered and authenticated users. Drupal allows setting up any number of custom user roles.

Besides the administrator, anonymous, and authenticated users, at the time of data collection (December 2015), Titan.uio.no defines the following user

\(^\text{1}\) https://www.drupal.org/project/usage/drupal
types that will be the target group of this thesis. These roles are staffed mainly by various faculty members who are ordinary internet users that have no expertise in Semantic Web technologies, but are at average intermediate users of Drupal’s content creation interface:

1. Blogger. Bloggers may access the content page, view the published content and own unpublished content. They can create new blog entries, edit, delete, publish and unpublish own blog entries, view unpublished blog entries, edit their wiki pages, and use the administration toolbar.

2. Author. Authors may access the content page, view published and unpublished content. They are allowed to create new articles, as well as editing, deleting, publishing and unpublishing own articles. They can create new factoids, edit and delete own factoids, edit wiki pages and use the administrator toolbar.

3. Editor. Editors may see the administrative control panel - dashboard, administer content and access the content page. They may create blog posts, articles, short articles and factoids. They can edit all the instances of the aforementioned content types they own, and delete any instance of these content types. Furthermore they can edit all wiki pages, publish and unpublish editable content and view unpublished blogs and articles.

In the context of allowing the aforementioned users to perform semantic enrichment, a central objective is providing a module compatible with their mental model, which presents a friendly user interface. Hence, consistency, simplicity, providing explanations, facilitating data entry, providing feedback and error recovery, and not causing a cognitive overload are some of the main challenges. The module must presume that the users lack expertise in semantic web technologies, and thus, ease learning. Moreover, fine grained access control should be implemented. One might hypothesize that users do not wish to burden themselves with learning to use a new module or read documentation and manuals; a hypothesis explored with the proposed research methods. Additionally, the degree of motivation varies between the user groups. Editors are trusted with expert tools and display a high grade of motivation as this is their full time job. Bloggers however may create content once a week and might not be as dedicated. To promote long term engagement and enforce access control, a Gamification-like strategy will be implemented.
1.2.3.1 Necessity of the module

Enabling non-experts to semantically enrich content is a necessity in this use case. Enrichment by experts can only be employed when the content is relatively static, for example in governmental websites. In this case, and all similar cases where new content is being served every day (see page 105), even an automated approach is not enough, since each instance of content has unique features that need to be tagged individually by field specialists.

1.3 Research Questions

The aim of this thesis is to explore how a CMS powered news outlet can perform Semantic Enrichment and what the results of this are with regards to visibility, content attribution, and a better SEO and Social Media strategy. In this context, the following research questions will be answered:

1. What are the requirements for developing an understandable module that will enable Semantic Enrichment for users with no expertise in the field of the Semantic Web?
2. What are the technical challenges faced?
3. What impedes the adoption of Semantic Enrichment by non-experts?

1.4 Activity Agenda

The following tasks will be carried out:

- Assess and test existing modules.
- Employ the methods and methodologies to elicit the requirements.
- Develop a high fidelity prototype.
- Assess design and user experience by conducting usability testing and redesign depending on the feedback.
- Conduct user acceptance testing to make sure that the module fulfills the user needs.

1.5 Outline

The thesis consists of the following chapters:

Chapter 2, Background: presents a non-exhaustive description of the technologies that underpin the Semantic Web, with emphasis placed on those relevant to the thesis. Furthermore a short introduction to Drupal and some relevant core and contributed modules is provided.
Chapter 3, Methods and Methodology: outlines the methods and methodologies used for data gathering and analysis.

Chapter 4, Design Guidelines and Prototyping: presents a list of design guidelines from the literature, and enumerates the ones that have been employed in the final module.

Chapter 5, Implementation: examines the technical part of the thesis; how this module was developed and how it can be used. The chapter also identifies the faults and shortcomings with the original module and explains how the developed solution aims to rectify these.

Chapter 6, Findings: provides the findings from applying the methods and methodologies described on Chapter 3.

Chapter 7, Discussion: evaluates the answers to the research questions and assesses the usefulness of the module. Moreover, it presents the challenges met and the reflections about the process. Finally, it provides suggestions for next step improvements.

Chapter 8: Conclusion: summarizes the work that has been executed.
2. Background

This chapter describes the concept of the Semantic Web and the building blocks that underpin it. In the second part, Drupal, the content management system that powers the target website is introduced. Core and contributed modules of interest are also outlined.

2.1 The Semantic Web Stack

Semantic Web technologies are made available as open standards by the World Wide Web Consortium (W3C). Development takes place in a step wise fashion where each layer is constructed on top of another. Two main principles hold (Alam, Rahman, Khusro, & Ali, 2015): downward compatibility and upward partial understandability. Generally the stack can be divided into four layers. The base layer consists of URIs, IRIs, XML and XML Schema, and Namespaces. The middle layer serves the implementation of core Semantic Web technologies. It consists of RDF, RDF Schema, ontology languages, and query languages. The layer on top consists of logic, trust, proof and cryptography, which are not standardized but provide enhancements to the lower layers by allowing the writing of application specific declarative knowledge, representation of proofs and proof validation, and gaining users’ trust for its operations and the information provided. The highest layer is the user interface and applications through which the users interact with the Semantic Web. This section provides a brief description of most of these layers, but delves into the ones relevant to the thesis.

Figure 1: The Semantic Web stack

http://www.w3.org/2006/Talks/1023-sb-W3CTechSemWeb/#%2819%29

---

2 http://www.w3.org/2006/Talks/1023-sb-W3CTechSemWeb/##%2819%29
2.1.1 RDF

Semantic Web applications require a flexible and domain independent data model. RDF (Resource Description Framework) is W3C's recommendation for a standard model for data interchange on the Web.

RDF represents information as statements consisting of an object, an attribute and a value. In the field's jargon these statements are called triples and the respective parts are called subject, predicate, and object (Hebeler, Fisher, Blace, & Perez-Lopez, 2011). This provides a seemingly rudimentary but powerful representation language for URIs (Uniform Resource Identifiers). A triple can be visualized as a graph similar to a directed labeled graph, where it is allowed to have more than one unique edges/predicates between the same pair of nodes, which are not necessarily connected to each other, and may form circles. The subject nodes contain resources, while the object nodes either resources or literals. The latter cannot be subject to further processing by parsers (Konstantinou et al., 2010), and may be strings, integers, dates etc.

RDF graphs are expressive, and easily understood by humans, but too abstract for information exchange between machines. Hebeler et al. (2011) enumerate some of the available serializations for converting the abstract RDF graph into concise forms such as byte streams or files. Among these serializations are RDF/XML, Turtle (Terse RDF Triple Language), N-Triples and N3. RDF's expressiveness is enriched by features such as blank nodes, named graphs, reification and collections.

2.1.2 URIs

URIs (Uniform Resource Identifiers) work as a unique global identification scheme for resources and are hence fundamental to the Semantic Web venture. This eliminates naming conflicts, ensures that two items are the same, allows combinations of data from distributed sources, and provides a path to additional information through reasoning (Hebeler et al., 2011). URIs establish an expansive namespace as they may contain URLs or URNs. It must be noted however, that URIs do not empower a semantically unambiguous interpretation of all RDF encoded information (Hitzler et al., 2011, p. 33). It is still possible to utilize different URIs for the same resource, just as it is possible to use the same URI for different things. Some of the RDF serializations offer a mechanism for abbreviating URIs using namespaces, into the form prefix:name, also known as QNames.
2.1.3 Ontologies

Ontologies are conceptualizations established between agents to assist the interpretation of the exchanged information. They consist of statements that define concepts, hierarchical relationships, and constraints. Besides exchange and interpretation of information, agents may infer new information by applying and extending the logical rules contained in the ontology.

Ontologies form an information domain model and can have varying expressive power. Some compare ontologies with database schemas or object-oriented class diagrams (Hebeler et al., 2011). A more accurate depiction identifies five layers within an ontology (Harris, 2004). Firstly, an ontology must be defined using a formal specification language. Secondly, an ontology specifies data structures. These implement only a part of the conceptual model. An ontology contains a set of assertions and integrity/inference constraints which define rules concerning the relationships between data structures. Some ontologies specify reference data in the form of constrained vocabularies or taxonomies. Reference data is used as components and classifiers of the instance data, which forms the final layer that agents exchange.

Non-trivial ontologies may still suffer from the possibility of logical inconsistencies (Alesso & Smith, 2008, p. 71). This does not negate the usefulness of ontologies, but points out their limitations. In contrast to ontologies, controlled vocabularies are simpler collections of concepts and terms used to describe a field of interest or area of concern. In both cases, reuse is a good practice as it “maximizes the probability that data can be consumed by applications that may be tuned to well-known vocabularies, without requiring further pre-processing of the data or modification of the application” (Heath & Bizer, 2011). However, there will always be a case where new terms need to be developed to describe aspects of a particular data set. In this circumstance, these terms should be mapped to related terms in well-established vocabularies.

2.1.3.1 Ontology Languages and frameworks

RDF is a generic, abstract model for describing resources using triples consisting of a subject, a predicate and an object. It provides however no domain-specific terms for describing classes of things and how to relate to each other. This function is served by ontologies, taxonomies and vocabularies expressed in RDF Schema, OWL, and SKOS (Heath & Bizer, 2011).
RDF Schema is a W3C recommendation that extends the basic RDF specification in order to provide generic language constructs for user-defined vocabularies - it does not introduce a topic-specific vocabulary for a particular domain. It has provided a minimal ontology representation language widely adopted by the research community (Shadbolt et al., 2006). In order to establish semantics, RDFS defines classes, properties, class hierarchies, property hierarchies and inheritance (Antoniou, Groth, Harmelen, & Hoekstra, 2011).

The expressive power of RDF and RDFS is deliberately very limited. OWL (Web Ontology Language) has been since 2004 W3C's recommended standard for the modeling of ontologies. Central to its design was finding a reasonable balance between expressivity and efficiency, as complex language constructs usually yield undecidability of reasoning (Hitzler et al., 2011, p. 111). This is why there exist three different sub languages (Full, DL, and Lite). OWL2, the forthcoming standard, builds on RDF and RDFS and uses an extension of their syntax. It allows for a more precise description of the conditions under which an instance belongs to a class, inference of relations between the classes, expressing equivalence, intersection, disjointness, difference and boolean combinations between classes, setting the local scope of properties, and defining special characteristics of properties such as symmetry, reflexivity, transitivity, uniqueness, inversion and cardinality restrictions (Antoniou et al., 2011, pp. 58 - 60).

2.1.3.2 Vocabularies

2.1.3.2.1 Schema.org

Schema.org is an initiative driven by major web search engines such as Google, Bing, Yahoo!, and Yandex. The vocabulary can be used with RDFa Lite 1.1, Microdata and JSON-LD. Schema.org is an attempt to define a broad, web-scale, shared, and easily extensible vocabulary focusing on popular concepts, since it is unrealistic for the aforementioned search engines to support every vocabulary in use. It is described as a “middle ontology that does not attempt to have the scope of an ontology of everything, or go into depth in one area. A central goal of having such a broad schema in one place is to simplify things for mass adoption and cover the most common use cases” (Ronallo, 2012).

Over 10 million sites use Schema.org to markup their web pages and email messages. The schema definition is maintained in a community-driven process, where prospective changes are announced and discussed in public
mailing lists and GitHub. Since its conception in 2011, schema.org has undergone more than 25 revisions, ranging from small typographical errors in the elements to the integration of entirely new vocabularies such as the Music Ontology, or Good Relations. Additionally, elements whose use is no longer encouraged are occasionally marked as deprecated, or are superseded by others (Meusel, Bizer, & Paulheim, 2015). It is however clear that the vocabulary has a bias towards search engine and commercial use cases (Ronallo, 2012).

2.1.3.2.2 Open Graph Protocol

Open Graph is a platform that enables turning web pages into social objects and capturing them in a social graph. Even though developed and promoted by Facebook, OGP is kept as generic as possible. It includes features such as publisher plugins and a simple RDFa-based markup.

OGP essentially “moved Facebook one step toward being a global linked data network” (Allemang & Hendler, 2011). This statement manifests as the ‘like’ mechanism is extended in order to include pages that were not already part of the network. Perhaps even more importantly, the markup helps Facebook connect the users across common interests and across different websites. What OGP has done could make vast parts of the consumer Web including movies, books, music, events, sports and news, semantically tagged. Hence, publishers have a stronger incentive to mark their content and get return traffic from Facebook. OGP is widely used by IMDb, Microsoft, NHL, Posterous, Pandora, Rotten Tomatoes, New York Times, Yelp and others (Yu, 2014).

For any web page to become a rich object in Facebook’s social graph it must be marked with extensible Open Graph tags. Without tags, “the Facebook Crawler uses internal heuristics to make a best guess about the title, description, and preview image for the content” (“A Guide to Sharing for Webmasters,”). The markup requires publishers to include at least 4 metadata properties in each object: title, type, image, URL. Placing objects into categories gives rise to various applications, for example personalized recommendations (MacManus, 2010). For Facebook to parse a page correctly, the OGP namespaces should be added to the <head>. The namespace changes depending on the object (articles, website, book, movie, custom etc.) that will be used. Each object provides access to unique properties. Code samples are provided below:

```html
<head prefix="og: http://ogp.me/ns# fb: http://ogp.me/ns/fb# website: http://ogp.me/ns/website#">""
```
Facebook Open Graph META Tags

Adding an AppID or a User ID gives access to Domain Insights, Facebook’s service that shows all referral traffic to a domain from Facebook. Domain Insights also provides sharing metrics and demographic information per domain and per URL so that content can be optimized for sharing, and tailored to the target audience.

OGP poses a few limitations, which are however justified design decisions as the markup was meant to remain simple, and it is believed that OGP’s abbreviated RDFa will actually help the adoption of the uppercase RDFa (MacManus, 2010) as more people will be consuming it. Moreover, RIF (Rule Interchange Format), which is a W3C standard, can convert Open Graph code into RDFa if required (MacManus, 2010).

1. The abbreviated RDFa allows using literals instead of URIs to identify classes. This format does leave room for ambiguity, and shifts away from the goal of Uppercase Semantic markup technologies which is to precisely refer to entities.

2. OGP does not promote the “Don't Repeat Yourself” (DRY) pattern which RDFa enables. It asks the developer to reiterate information which is likely to already exist in the page.

3. OGP redefines vocabulary terms which have been around for many years. Thus, existing RDF data which might have already be using legacy vocabularies now need to add OGP’s specific terms to be included in the Open Graph. RDF datasets ending up with duplicate terms for the same semantics is a recurrent problem every time a new big player adopts RDF. Such terms are: og:image - foaf:depiction, og:latitude - geo:lat, og:postal-code - vcard:postal-code, og:email - foaf:mbox and og:phone_number - foaf:phone.

4. Only one object can be specified per web page. This prevents multiple topics or people on a single web page from being marked up. OGP does not disambiguate a webpage and all the resources it might describe. In OGP’s eyes, the social objects are the HTML documents and not the real concepts or physical objects and people, that people are likely to show an interest in. An example would be wanting to like a particular comment on a page and
not the whole page, or a page about a music album and all the songs it contains.

5. OGP introduces og:type, an alternative to rdf:type. The rationale behind it is to keep the markup consistent with the <@property> and <@content> syntax. However, because the @content attribute is used, it means that a string is required as the type of the object. This poses a limitation in OGP: it is not possible to specify several types for the same object, for example it is not possible to express that a person is both an actor and a director, something which would easily be specified using RDFa's typeof attribute if only URIs were used instead of strings. Typeof disposes of the single type limitation, and RDF classes which look like strings, can be used thanks to the CURIE syntax.

2.1.3.2.3 Twitter Cards

Twitter Cards allow website owners to add markup into their page content, so that when users are presented with content previews, images, video and more information extending the normal 140-character limit. This does not only mean that publishers can control how their content is displayed in tweets, but it also drives traffic to the website (Yu, 2014) and increases the number of followers through content attribution.

The types of cards currently supported are³: the Summary Card that contains a title, a description, a thumbnail and a Twitter account attribution, a Summary Card with a large image, an App Card which details a mobile app with direct download, and finally a Player Card to provide multimedia (Cards for photos, galleries, and products have been deprecated). The Twitter Card processor first looks for markup using Twitter properties, and if none are found, it defaults to the supported OG properties (Yu, 2014). A code sample is provided below:

```xml
<meta name="twitter:card" content="summary" />  
<meta name="twitter:site" content="@flickr" />  
<meta name="twitter:title" content="Small Island Developing States Submission" />  
<meta name="twitter:description" content="View the album on Flickr." />  
<meta name="twitter:image" content="https://farm6.staticflickr.com/5510/12f_z.jpg" />  
```

Twitter Card Analytics is a framework that measures performance, gives related insights into how the content is being shared, and reveals how key

---

³ https://dev.twitter.com/cards/types
metrics such as URL clicks, app install attempts and Retweets can be improved.

It should be noted that Twitter Cards do not follow any of the W3C standards, or introduce/use any ontologies and declare a namespace. Twitter Cards only utilizes simple property-value pairs. Additionally, only one card type may be specified per content page, while there is no such concept as a URI. It can be thus debated that Twitter Cards is not a semantic technology yet. In any case, structured data is better than the lack of thereof.

2.1.3.2.4 Dublin Core

The Dublin Core vocabulary defines metadata elements that facilitate the classification, connection, and discovery of document-like objects (videos, images, web pages), but also physical creative objects (books, academic articles, artwork). The 15 fundamental terms described below constitute the Dublin Core Metadata Element Set 1.1, and belong in the /elements/1.1/ namespace. A fuller version is defined in the Dublin Core Metadata Initiative (DCMI) set, which contains even more elements such as audience, bibliographicCitation, dateCopyrighted, dateSubmitted, educationLevel and so forth. These reside in the /terms/ namespace. The vocabulary is endorsed in several international standard documents.

Table 1 Dublin Core Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Element Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creator</td>
<td>the person or organization responsible for creating the content of the resource, e.g., authors in the case of written documents</td>
</tr>
<tr>
<td>Publisher</td>
<td>the entity responsible for making the resource available in its present form; it can be a publishing house, a university department, etc.</td>
</tr>
<tr>
<td>Contributor</td>
<td>the person or organization not specified in a Creator element who has made significant intellectual contributions to the resource but whose contribution is secondary to any person or organization specified in a Creator element, e.g., editor, transcriber, illustrator</td>
</tr>
<tr>
<td>Title</td>
<td>the name given to the resource</td>
</tr>
<tr>
<td>Subject</td>
<td>the topic of the resource. Normally this will be expressed as keywords or phrases that describe the subject or content of the resource</td>
</tr>
<tr>
<td>Date</td>
<td>the date associated with the creation or availability of the resource</td>
</tr>
<tr>
<td>Identifier</td>
<td>a string or number that uniquely identifies the resource. Examples include URLs, Purls and ISBN, or other formal names</td>
</tr>
<tr>
<td>Description</td>
<td>a free text description of the content of the resource, and has flexible format, including abstracts or other content descriptions</td>
</tr>
</tbody>
</table>
This section lists a few prevalent actors and the vocabularies they have launched. The list is not exhaustive.

Google's technologies: Google has launched various technologies promoting semantic enrichment. Among these are the Rich Snippets, the Knowledge Graph, and the GMail markup.

Snippets are small samples of the content of a website displayed in Google's search results. Rich Snippets (Goel, Gupta, & Hansson, 2009) apply Google's algorithms to highlight structured data embedded in web pages. Users are presented with brief information about their search results at a glance. Various data types are supported: reviews, ratings, people, times, recipes and so forth. To display Rich Snippets Google employs Microformats and RDFa.

The Knowledge Graph is a semantic knowledge base aimed at enhancing traditional search engine result pages with information gathered from a variety of sources. These sources can be pages indexed by Google, objects on Maps, public data sources such as Wikipedia, LOD datasets such as DBpedia, Freebase, the FDA datasets, the Weather Underground or the World bank (Sikos, 2015). The Knowledge Graph contains more than half a billion objects and more than 18 billion facts about relationships between different objects. As mentioned in Google's research blog, the Knowledge Graph is expected to stir the feature of search: “Now, with the Knowledge Graph, we are beginning to think in terms of entities and relations rather than keywords” (Gillick & Orr, 2014). When the publisher of some content is
the authority of it, Google treats structured data as factual and imports it into the Graph. Schema.org and Microdata or JSON-LD may be used.

Finally, GMail markup employs Schema.org to accelerate user actions from GMail’s interface and highlight important information. GMail markup supports various types of actions and interactive cards. The concept is to put “Linked data at the center of getting computing and the monotonous details of our everyday grind out of the way” (Sporny, 2013b).

FOAF: FOAF (Friend of a friend) is an ontology describing people, their activities and their relations to other people and objects. The ontology is primarily organized around the following classes: foaf:Person, foaf:Group, and foaf:Document. It is extensible, distributed, and operates under the AAA (Anyone can say Anything about Any topic) principle (Allemang & Hendler, 2011). FOAF has been gradually evolving since its creation in mid-2000. The core classes are stable and will not be changed, yet new terms might be added. Besides the core, the main FOAF terms can be categorized as social web terms and linked data utilities (Brickley & Miller, 2014). The vocabulary is identified by the namespace URI http://xmlns.com/foaf/0.1/. According to Heitmann, Cyganiak, Hayes, and Decker (2012) who studied 124 applications that were submitted to the Semantic Web challenge during the 2003-2009 timespan, FOAF (27%) was proven to be the most popular vocabulary followed by DC (15%) and SIOC (7%).

SIOC: Posts, user roles, threads, user accounts, and user groups of online communities can be described using Semantically-Interlinked Online Communities (SIOC). The namespace of SIOC Core is http://rdfs.org/sioc/ns#.

DBPedia: DBpedia is a crowd-sourced community effort that extracts structured, multilingual knowledge from Wikipedia and makes it freely available by utilizing Semantic Web and Linked Data technologies. The latest report (Lehmann et al., 2014) documents that the project consists of over 400 million facts that describe 3.7 million things in the English edition, while the knowledge bases from the rest 110 editions consist of 1.46 billion facts and describe 10 million additional concepts. DBpedia has 27 million RDF links to more than 30 external data sources, while several hundred data sets point to DBpedia, making it one of the central interlinking hubs in the LOD cloud.
The extraction framework is the technical core of the project. Wikipedia articles consist principally of free text, but also of various types of structured information in the form of wiki markup. The framework parses the Wikipedia pages into an Abstract Syntax Tree which then is forwarded to the extractor, which yields a set of RDF statements. DBpedia maintains a single, shared, community-curated ontology consisting of 320 classes, which form a subsumption hierarchy, and 1650 properties. There exist mapping communities for 27 languages, 23 of which are active. The main namespaces used are: http://dbpedia.org/resource/ (prefix dbr) for representing article data, http://dbpedia.org/property/ (prefix dbp) for representing properties, and http://dbpedia.org/ontology/ (prefix dbo) for representing the DBpedia ontology. Since internationalization is a central element of the project, the mappings in the ontology are created through a worldwide crowdsourcing effort. The DBpedia Mapping Wiki maps templates from all the different language versions of Wikipedia into the shared DBpedia ontology. The DBpedia Live system processes a continuous stream from Wikipedia so that data does not become outdated. Not only are the knowledge bases downloadable, but a number of SPARQL endpoints are also provided.

Rich Pins: Rich Pins, launched by Pinterest allows publishers to attach structured data to their content so that the generated pins provide more relevant information. Six types of pins are currently supported\textsuperscript{5} : App, Place, Article, Product, Recipe and Movie. To create a rich pin, both Schema.org and OGP may be used.

2.1.3.2.5 Choosing Vocabularies

Even though there is no definitive directory to consult when in need of a suitable vocabulary or ontology, there are some useful starting points. Heath and Bizer (2011) suggest SchemaWeb, SchemaCache and Swoogle, as well as consulting patterns and levels of vocabulary usage ‘in the wild’. They also propose a series of criteria when selecting vocabularies for reuse. A desirable candidate vocabulary should have widespread usage and uptake. Active maintenance and a clear governance process is another requirement. The vocabulary should also cover enough of the data set and the degree of expressivity should be appropriate for the scenario in question.

2.1.4 Semantic Annotation Formats

The Web is primarily made up of HTML documents, that do contain significant amounts of structured data (Adida, Birbeck, McCarron, &

\textsuperscript{5}https://business.pinterest.com/en/rich-pins
Herman, 2015) but remain largely unavailable to tools and applications. When publishers are able to express this data more completely and when tools can read it, new functionality emerges. Users may transfer structured data between web sites and applications, and browsers can improve the user experience: an event from a web page can be directly imported into a user’s calendar; a license on a document can be detected so that users can be informed about their rights automatically, a picture’s creator, resolution and location can be easily published, enabling structured search and sharing.

Structured data can be annotated in the markup, or written in separate machine readable metadata files. The annotation languages extend the core XHTML markup with additional elements and attributes through external vocabularies that contain the terminology and properties of a knowledge representation domain, as well as the relationships between the properties in a machine readable form.

2.1.4.1 RDFa

RDFa (RDF in attributes) is an official W3C Recommendation (Adida et al., 2015) for expressing RDF triples in (X)HTML, XML or SVG as attribute values (Sikos, 2015). RDFa Core is the full specification that can convey sophisticated annotations, while RDFa Lite is a minimal subset, easier to learn and suitable for more general scenarios.

Among the attributes that extend the standard XHTML tags are: vocab, typeof, property, resource, prefix, about, content, datatype, inlist, rel, rev, href, src, and xmlns (Segaran, Evans, & Taylor, 2009). A code sample follows:

```
<div vocab="http://schema.org/" typeof="Person">
  <span property="name">John Doe</span>
  <img src="JohnDoe.png" alt="John Doe" property="image"/>
  John's web site: <a href="http://www.johndoe.com" property="url">johndoe.com</a>
</div>
```

RDFa uses a graph to model the Web, unlike the other approaches that use trees. Items and item properties can be identified by IRIs, mappings to RDF are well specified, and prefix rebinding as well as vocabulary mashups are allowed (Sporny, 2011). The most frequently used vocabulary namespaces (cc, dc, dcterms, foaf, og, sioc, and schema) are predefined in parsers, so they may be omitted (Sikos, 2015). RDFa also provides a Document Object Model (DOM) API to extract and utilize structured data from pages (Sikos, 2015).
RDFa Lite is feature-equivalent to Microdata, and fully upward-compatible with RDFa 1.1 allowing seamless migration (Sporny, 2012).

RDFa is supported by all the major search engine crawlers (Sporny, 2012) and using it to express rich snippets results in the pages being “displayed in an enhanced format” (“Linked Data in HTML,”). Additionally, since RDFa has been employed by Facebook’s Open Graph and integrated in CMSs such as Drupal, it is pointed to “as the de facto standard” (De Nart, Tasso, & Degl’Innocenti, 2014). The aforementioned facts, together with the reported broader deployment of RDFa (Sporny, 2012) justify why this format is chosen as a W3C standard over the other two approaches. Since it is the most expressive, “implementing RDFa has proven to be overly complex for most Web developers. As a result Google […] has discovered a large error rate in the application of RDFa by webmasters” (Ronallo, 2012).

### 2.1.4.2 Microdata

Microdata is a separately specified HTML5 extension, adding attributes to the core vocabulary to represent structured data as a group of name-value pairs. These groups are called items and each name-value pair is a property. Items and properties are created by the `itemscope` and `itemprop` attributes respectively. The type of the items and item properties are expressed using the `itemtype` attribute, by declaring the web address of the external vocabulary that defines the corresponding item and properties (Sikos, 2015).

A code sample follows:

```html
<div itemscope="itemscope" itemtype="http://schema.org/Person">
  <span itemprop="name">John Doe</span>
  <img src="JohnDoe.jpg" alt="John Doe" itemprop="image"/>
  John's web site:
  <a href="http://www.johndoe.com" itemprop="url">johndoe.com</a>
</div>
```

Even though Microdata boasts a high adoption rate (Sporny, 2013a) and a simpler syntax to that of RDFa (Sporny, 2011), it has also been subjected to criticism (Sporny, 2012). Support for the Microdata API has been removed both from Safari and Chrome (Sporny, 2013a) and from the main HTML 5.0 specification (Cotton, 2013) keeping it as a separate specification which will be further developed only if “editorial resources can be found”. The fact that Microdata lacks necessities such as an active community, a specification editor and a solid test suite, is a major factor of concern for the future (Sporny, 2013a). Schema.org may have started with focusing on Microdata, but they now support multiple syntaxes such as RDFa and JSON-LD.
(Brickley, 2012) since “there are certain things that are much harder in Microdata, like mixing vocabularies, inverting the direction of a property relationship, providing clear mappings to RDF, allowing multiple elements per statement and using compact IRIs” (2015; Sporny, 2011). Microdata is supported in Google’s markup, and outputted from the module developed in the context of this thesis.

2.1.4.3 Microformats

Microformats publish structured data about basic concepts such as people, places, events and multimedia by applying and reusing features of existing technologies, such as the class, rel and rev attributes of (X)HTML, while others extend the core markup based on Plain Old Semantic HTML (Sikos, 2015). Over time, several types of microformats (hCalendar, hCard, hMedia, hNews, hProduct, hRecipe, hResume, hReview) have been developed, however only hCard and hCalendar have been ratified (Baofu, 2015). A code sample marked up with the hCard microformat follows:

```html
<link rel="profile" href="http://microformats.org/profile/hcard" />
...
<div class="vcard">
  <span class="fn">John Doe</span>
  <img class="photo" src="JohnDoe.jpg" />
  John's web site: <a class="url" href="http://www.johndoe.com">johndoe.com</a>
</div>
```

Microformats are “designed for humans first and machines second”⁶, and lower the entry barrier due to the “reduce, reuse, recycle” (Khare, 2006) design principles they are based upon. However, “due to limitations and open issues, other machine-readable annotation formats gradually overtook” (Sikos, 2015). Simplicity in this case comes at the cost of limited application areas and extensibility (Hitzler et al., 2011). To start with, Microformats do not provide URI and IRI support since each microformat uses its own predefined vocabulary. Thus, unambiguous representation is hindered. It is also impossible to express overlapping objects. Moreover, vocabulary mashups are not implemented (Sporny, 2011), and the publisher cannot apply various microformats as multiple values on the same element. Locally scoped vocabulary terms and item chaining, an important property when expressing social networks, are not supported either. In addition there is no mechanism to serialize data into JSON, no DOM API implementation, and no unified parser specification to extract structured data (Sporny, 2011). Microformats

⁶ http://microformats.org/about
are not related to the W3C, and there is no standard mapping to RDF (Sporny, 2011). Finally, newer HTML5 structural elements are not recognized by all parsers (Sikos, 2015). Additionally, no new microformats have been released, while the mailing list has been inactive since 2012.7

2.1.4.4 JSON-LD

JSON-LD (JavaScript Object Notation - Linked Data) is the newest W3C standard (Sporny, Kellogg, Lanthaler, & Group, 2013) based on the JSON data serialization and messaging format. It allows the author to embed a block of JSON data inside a script tag anywhere in the HTML code. In contrast to RDFa and Microdata, the JSON-LD code is completely separate from the (X)HTML code. This results into the semantic data not being interleaved with user-visible text, which allows for more expressiveness8 and better readability (Sikos, 2015). The syntax is designed to easily integrate into deployed systems that already use JSON, providing a smooth upgrade path. It aims to release Linked Data in Web-Based programming environments, build interoperable services, and store Linked Data in JSON based storage engines like MongoDB. JSON-LD is fully compatible with JSON, so the multitude of existing parsers and libraries can be reused. Additionally, JSON-LD introduces:

- a universal identifier mechanism for JSON objects through the use of IRIs, and key disambiguation,
- a mechanism in which a value in a JSON object may refer to a JSON object on a different site,
- the ability to associate data types with values such as dates and times,
- a facility to express one or more directed graphs such as social networks, in a single document.

JSON-LD is also a complete RDF syntax, extending the data model as follows:

1. JSON-LD properties can either be IRIs or blank nodes, whereas RDF predicates must be IRIs. RDF has been criticized (Sporny, 2014) for not being as elegant a data model as proclaimed, since blank nodes are applied inconsistently and native graph support does not exist, ailments which JSON-LD attempts to cure through blank node support. There are also some predicates, such as JSON keys, that are local to the document, and blank nodes are necessary in this case.

7 http://microformats.org/discuss/mail/microformats-dev/
8 https://developers.google.com/search/docs/guides/intro-structured-data?hl=en
2. In JSON-LD lists are a first-class construct whereas in RDF they are part of the RDFS vocabulary.

3. RDFS values are either typed literals or language tagged strings. JSON-LD supports native JSON datatypes and booleans accompanied with conversion rules.

A code sample follows:
```html
<script type="application/ld+json">
{
  "@context": "http://schema.org",
  "@type": "Person",
  "image": "JohnDoe.jpg",
  "name": "John Doe",
  "url": "http://www.johndoe.com"
}
</script>
```

As stated in the standard specification (Sporny et al., 2013), JSON-LD satisfies design goals such as simplicity, easy learning curve, seamless compatibility, syntactic expressiveness, terseness and smooth and simple transition from existing JSON based system. To work with RDF one typically needs a quad store, a SPARQL engine and libraries. However “a standard web developer has no interest in that toolchain because it adds more complexity to the solution than necessary” (Sporny, 2014). Employing a technology that most web developers already use helps with solving interesting distributed problems “without buying into any grand vision” (Sporny, 2014).

### 2.1.4.5 Adoption

The most recent analyses (Bizer et al., 2013; Meusel et al., 2015; Meusel, Petrovski, & Bizer, 2014) regarding the adoption of different markup languages, the main topical areas of the published data, as well as the different vocabularies used, are based on the Common Crawl web corpora. The Common Crawl foundation issues publicly available, large-scale crawls covering billions of pages and the compressed size of the resulting corpus comes up to tens of terabytes. Out of these crawls, the Web Data Commons project regularly extracts structured data. The crawler used by the Common Crawl Foundation relies on the PageRank algorithm for deciding which pages to retrieve. Using PageRank results in the corpus being “a snapshot of the popular part of the Web. On the other hand, it also results in the number of pages that are crawled per website to vary widely. For instance, youtube.com is represented by 93.1 million pages, whereas 37.5 million PLDs (Pay Level Domains) are represented by less than 100 pages” (Bizer et al., 2013). Even
though proper representation is an issue and the results are influenced by the crawling strategies chosen, using publicly available data “allows for a new methodology of analyzing standard adoption, where […] we can observe the adoption directly from the data, published by hundreds of thousands of standard adopters” (Meusel et al., 2015). Additionally, it becomes possible to produce scientifically verifiable results (Bizer et al., 2013). When deployment by popularity is a matter of interest, the Alexa lists were consulted. Alexa Internet Inc. maintain a ranking of the most frequently visited websites. Varying amounts of dataset series dating from 2010 to 2015 are studied. Hence, it is possible to get an insight on how the markup formats evolve. Despite the differences all analyses agree that Microformats is the most widely adopted technology. There is no unanimity when it comes to RDFa versus Microdata, as the two older studies point at RDFa, while the most recent crawl indicates that Microdata is the next most popular format. Since JSON-LD is a rather recent standard no data is published yet. The analyses converge regarding the topical areas where each markup language is most popular, as well as the prevailing vocabularies and classes for each of the markup formats.

2.1.4.6 Upper versus Lowercase Semantic Web

The term “Uppercase Semantic Web” has been coined to pinpoint research efforts attempting to bring full-fledged reasoning capabilities to intelligent software through formal, standardized semantic technologies, while the “lowercase semantic web” aims at encoding semantic data that can be accessed by everyday software and created by unsophisticated users (Di Iorio et al., 2012). The two approaches are not in competition, but rather the latter is an intermediate step for the former (Murugesan, 2009, p. 348). There have also been advances in reconciling both developments, such as RDFa and GRDDL (Hitzler et al., 2011). According to its creators, JSON-LD “intends to marry the lowercase semantic web with the uppercase Semantic Web in a way that was friendly to developers. For developers that didn’t care about the uppercase Semantic Web, JSON-LD would still provide a very useful data structure to program against” (Sporny, 2013c).

2.1.4.7 Metadata Standards

Metadata Standards represent the best practices for metadata interoperability (Khosrow-Pour, Clarke, Jennex, Becker, & Anttiroiko, 2012, pp. 89-90). An example of metadata standard is the Dutch e-Government
Overheid.nl Web Metadata Standaard; OWMS\(^9\). OWMS builds on the Dublin Core Metadata Element Set. It defines a small number of core elements that must be present in the description of a page, namely its identifier, its title, its information type, its language and the date of last modification. Furthermore, either the creator or the organizational responsible must be defined. This information is typically embedded in the head of the HTML page, but XML and XHTML+RDFa syntax is also supported. While other similar initiatives exist both in European and non-European level, OWMS provides concise examples that have been used to guide the development of the module. For example, for the `dcterms:creator` attribute OWMS recognizes the following ways to create a statement as legal:

```xml
<meta name="DCTERMS.creator" scheme="" content="" />
<link rel="DCTERMS.creator" href="" />
<link rel="DCTERMS.creator" href="" title="" />
```

### 2.1.5 Publishing tools and storage approaches

In order to publish semantic data there exist various patterns to complement the underlying infrastructure. The primary consideration in selecting a workflow depends on the nature of the input data, but additional factors that have a bearing on the choice are the volume and the rate at which data changes (Heath & Bizer, 2011).

#### 2.1.5.1 Triplestores, SPARQL, RDB2RDF

A database manager optimized for RDF triples is called a triplestore (DuCharme, 2013). Some triplestores focus on providing a rich means to reason over the triples, while others focus on storing large quantities. Some operate as plug-ins to current Web browsers and others as systems that can work with a range of existing third-party databases. Some of the most commonly used frameworks that come with a native triple store are Virtuoso, Redland, Sesame and Allegrograph (Hitzler et al., 2011, p. 329).

SPARQL (Simple Protocol and RDF Query Language) is a W3C recommendation denoting a family of standards including a query language for RDF, a protocol definition for sending SPARQL queries from a client to a query processor and an XML serialization format for results returned by a query. The SPARQL syntax bears similarity to that of SQL (Konstantinou et al., 2010). Triplestores offer endpoints where SPARQL queries are submitted through HTTP.

\(^9\) [http://standaarden.overheid.nl/owms/terms](http://standaarden.overheid.nl/owms/terms)
A typical SPARQL query starts with prefix declarations (@PREFIX). Variables can occur anywhere in a query and are denoted with “?” or “$”. Four query forms are provided: SELECT, CONSTRUCT, ASK and DESCRIBE. Just like SQL, the SELECT FROM WHERE form is still present. The results are returned in a set of mappings called bindings. ASK checks whether a graph pattern exists in a data set. CONSTRUCT retrieves a whole RDF graph and is often used to translate between schemata (Antoniou et al., 2011, p. 52). DESCRIBE requests triples that describe a particular resource. It is up to the query processor to decide which triples are the description of a resource, and this has led to inconsistencies (DuCharme, 2013, p. 104). In order to combat the effects of the Open World principle, SPARQL employs the OPTIONAL, UNION and FILTER keywords and various functions. Results may be organized using the same keywords as in SQL. Nested queries as well as the IF, BIND and COALESCE functions are available (DuCharme, 2013).

Datasets stored in relational databases usually power important legacy applications, but can be published as semantic data thanks to RDB-to-RDF wrappers. These tools define mappings from RDB schemata to RDF graphs that are served on the Web and usually offer SPARQL endpoints, while respecting Linked Data principles. One widely used tool designed for this purpose is D2R server and D2RQ (Bizer & Cyganiak, 2006). In effort of providing a standard, vendor independent mapping language the W3C RDB2RDF Working Group recommends R2RML (Das, Sundara, & Cyganiak, 2012). The main drawback is that such solutions can only be used by developers or advanced users that possess the required knowledge. Linking a database to an ontology must involve a domain specialist as well. In these cases a mapping tool with a graphical interface can be indispensable.

### 2.1.5.2 APIs, Textual data

When having access to structured data behind a custom API, such as the Twitter Web API, a custom wrapper might have to be developed according to the API’s requirements, as each one provides heterogeneous query and data retrieval interfaces and returned results using different formats. The wrappers must initially assign HTTP URIs to the resources about which the API provides data. When one of these URIs is dereferenced, the wrapper rewrites the client’s request into a request against the underlying API. Then, the results of the API request must be transformed into RDF and sent back to the client (Heath & Bizer, 2011). Thus it can be trivial and effective to implement a wrapper from scratch.
When the input is textual documents in natural language, it is possible to pass the documents through an entity extractor like Open Calais\textsuperscript{10} or DBpedia Spotlight\textsuperscript{11}.

![Image](image.png)

Figure 2 Publishing and Storage

### 2.1.6 Logic, Rules, Inference

Logic, the study and application of the principles of reasoning (Alesso & Smith, 2008, p. 23) is used as the basis by which formal languages represent knowledge, convey semantics and act as automatic reasoners to provide proofs and infer conclusions. Description Logic is a subset of First Order Logic that has the advantage of decidability and tractability, but is limited to classification and subsumption (Alesso & Smith, 2008, p. 30). It is an important factor in the development of the Semantic Web as OWL DL and OWL Lite are based on it.

Logical rules establish the valid relationships between a set of premises and an assertion that will be the conclusion (Alesso & Smith, 2008, p. 151). Inference engines execute of a set of rules to process the knowledge available on the Semantic Web, deduce new facts and find inconsistencies, thus improving the quality of data integration (“Inference,” 2015). RIF (Rule Interchange Format) is a W3C recommendation for rule languages (Hebeler et al., 2011, p. 234). Rule engines are suitable mainly for mission critical production environments where concrete and robust solutions are required. In the rest of the cases, a set of if-then-else statements suffices (Antoniou et al., 2011).

\textsuperscript{10}http://www.opencalais.com/

\textsuperscript{11}https://dbpedia-spotlight.github.io/demo/
2.2 Drupal and the Semantic Web

This section reviews some of the core and contributed modules that deploy Semantic Web technologies in Drupal.

2.2.1 Core Modules

**RDF**, described in detail by Melançon et al. (2011), is a core Drupal 7 module that does not include any user interface but only provides an API for other modules to use much like the Field module. Many core modules including Node, Comment, User, Taxonomy, Forum, Blog and Tracker leverage the RDF module. It describes the data structure of a Drupal site in terms of RDF mappings, which relates a Drupal field to one or more RDF terms and then takes these mappings and inserts them into the HTML output in the form of RDFa attributes. The module utilizes the concepts of entities, bundles and fields introduced in Drupal 7.

Entity types have a set of attributes such as title, or date, which can be mapped to an RDF property. This RDF property can be added to the HTML. Aside from attributes of an entity type, a special kind of mapping exists for the actual type of an entity. This is not redundant from the viewpoint of the external applications that have no knowledge about the internal structure of the website. The RDF module allows the entity type to appear in the HTML thanks to the `rdftype` key of the RDF mapping. An RDF mapping is a nested associative array defining a relation between Drupal’s internal attributes and RDF predicates designed to be understood by and interoperate with external applications. A mapping structure contains 3 required keys. The `type` and `bundle` keys refer respectively to the entity type and the bundle to which the RDF mappings pertain. The third key, `mapping`, lists the Drupal attributes that should be mapped to RDF. Besides the special `rdftype` key, the other keys refer to either Drupal custom attributes or to fields defined by the core Field module. Each item contains an array listing the RDF predicates of this attribute the predicates key. When not dealing with strings, but with resources with a URI, the developer should specify a type for a mapping element.

The major vocabularies used by Drupal are DC, FOAF, SIOC and SKOS, but modules can still add new namespaces and their associated prefix by implementing `hook_rdf_namespaces()`.
2.2.2 Contributed Modules

The following are a few contributed projects that extend Drupal 7’s RDF capabilities. They all are actively maintained and under active development:

The Schema.org module allows site administrators to specify what schema.org terms they wish to associate with their content types and their fields. For each content type, the editor/publisher may navigate to the Schema.org tab and select a type and a title property. For each field, a Schema.org mapping may be selected. The module has been successfully tested to produce Rich Snippets on both live results and the Google Rich Snippets testing tool. 4443 websites have reported using the module and it has been downloaded 143381 times. While it was developed for Drupal 7, it is being updated to Drupal 8.

RDF Extensions offers a set of modules to directly interact with the core RDF module. Site administrators who prefer a user interface over writing code can use the RDF UI module to alter the RDF mappings. This package also offers more RDF serialization formats like RDF/XML, N-Triples and Turtle. Additionally, it is possible to import vocabularies that can be used in RDF UI. The project has been installed in 1710 sites and downloaded 18715 times.

The SPARQL module turns a Drupal website into a SPARQL endpoint by indexing all its RDF data. Site administrators can also register external endpoints that other modules can use to get their data from. The package exposes the API for other modules to submit SPARQL queries on the fly without setting up an endpoint locally. The project has been installed in 275 sites and downloaded 12982 times.

SPARQL Views is a query plug-in for Views allowing the retrieval of data from SPARQL endpoints into Views. 95 websites have reported using the module and it has been downloaded 5167 times.

Drupal 7 implements out of the box RDFa 1.0. The purpose of the RDFa module is to implement the new features of the simpler RDFa 1.1 Lite. The module supports Panels and in order to work with a UI it can be used in conjunction with the schema.org module. It has been installed in 195 sites and downloaded 954 times.

The Good Relations for Drupal Commerce module adds Good Relations markup on Drupal Commerce product displays, allowing the generation of Rich Snippets. The title, price, currency and ratings fields are currently
supported. The project has been installed in 970 sites and downloaded 2365 times.

Finally, the **JSON-LD** module outputs Drupal entities as JSON-LD through the RESTful Web Services. It currently requires the RDF Extensions module and utilizes the php-json-ld library that is automatically downloaded if the module is enabled via drush. In order to view the JSON-LD output, the .jsonld extension can be used after the entity URI. The project has been downloaded 1466 times.

### 2.2.2.1 Meta tags

Besides semantically annotating objects in a piece of content with vocabularies like Schema.org that can be deployed with RDFa Lite 1.1, Microdata and JSON-LD, meta elements may also be seen as a semantic technology. Meta tags are page level tags added to the head section of an HTML page and are a part of any SEO and Social Media strategy for a website (Williams, 2012). Even though practices like keyword stuffing are no longer a valid technique to achieve higher page ranking, the importance of meta tags is still present as they are a way for content providers to add extra information to a web page, typically for the benefit of search engines to learn more about the purpose and meaning of a website and allowing robots to validate content relativity and determine site niche (McMurray, 2012). For example the description meta tag is an important element despite the fact that it does not actually serve higher-ranking purposes. Well written, client oriented meta descriptions show up on the search results and may stimulate the interest of potential visitors. Page level tagging is easier than recognizing an tagging individual objects. Additionally, if vocabularies like OGP are to be used, page level tagging is the only option. This is because OGP allows only one object to be specified per page.

**Metatag** is the most widely deployed module that allows to automatically provide structured metadata about a website. It is marketed as a SEO module and during the time of writing (spring 2016) it has 1,715,451 downloads, is actively maintained and under active development, while a pre-release version for Drupal 8 exists. It depends on the CTools and Token modules. Some of its features include:

1. Support for a large number of commonly used meta tags, including page title, canonical URL description, OGP, Twitter Cards and Dublin Core.
2. Global settings to control meta tags on all pages
3. Per entity and per entity bundle support to control default meta tags for all items of a certain type, e.g. all nodes of a certain content type.
4. Ability to override meta tags on each entity object, for example on an individual node, from the edit page.
5. Multilingual support using the entity translation module and translation support using the i18n module.
6. Compatibility with the Revisioning and Workbench modules in order to support entity revisions and workflows based upon revision editing.
7. Individual permissions for each meta tag.
8. Meta tags for admin pages.
9. Exportable default meta tag configuration.
10. Extendable and flexible API to define own meta tags.
11. The fields can be populated using the data associated with the Token, so the users do not have to enter meta tag data.

The module presents however a few issues (Williams, 2012):

- It covers nodes, but lacks Views and Panels integration. One workaround is to make every page into a node and then add any view as a block under the node.
- It does support a wide range of meta tags, but not semantic markup with schema.org or any ontology of choice. It does not support custom Open Graph types.
- The edit feature has an all-or-nothing character. The administrator might want to give users access to edit their meta data but restrict their ability to edit it when they create a node. At this time, the level of permission granularity does not appear to be an option.
- Token browsing may not be straightforward for inexperienced users.

Other relevant but less used modules are:

The Meta tags quick module has the ability to assign meta values based on a path. Every meta tag (keywords, description, copyright) is treated as a separate field. Meta tag fields are then attached to registered entities (content types, taxonomy vocabularies) and appear in the entity edit form. It can be used to tackle the lack of views integration in the Metatag module. Once there is a view set up, the editors/publishers can navigate to the module’s settings and create a path-based meta tag for that view page. This will however add tabs to the views, and it might break the clean look of the page. Besides that, the meta tags quick module is built on the Fields API. In practice this requires setting up meta tags on each and every content type, when most of the time consistent meta attributes across all the content of the site are required. Removing the module takes some time too, since the fields it creates are left on the content types and have to be removed manually before the module can be uninstalled. On its own, the meta tags
quick module requires at least one more module to complete its meta features. Such a module would be Page Title, which adds the `<title>` tag to a page’s HTML. Finally, Meta tags quick does not come with Twitter support out of the box.

The Open Graph meta tags module adds tags to a node to enable it to become a rich social object. It also allows to easily select the image thumbnail used to represent the node on the preview constructed by Facebook. The editor/publisher is shown a list of thumbnails of all the images associated with the node, both as fields and as images embedded within the node’s body. The editor may set default values for OG meta tags across all nodes and then override them on a node by node basis. It is possible to restrict the use of OG meta tags to specific content types and restrict access to administering the module to specific roles. The installation requires that an XML namespace attribute is added to the HTML tag at the top of the `page.tpl.php` file. However, this module does not output the `fb:admins` or `fb:app_id` tags which are required for integrating with Facebook Insights.

The custom Meta module provides an interface for adding and managing meta tags. The editor/publisher simply selects a meta attribute (for example property) and then fills a form with the value for the defined attributes and the content. A list of all the tags is displayed in the box below these options. The created meta tags are added to the html head of all non-admin pages, and arguably, this module is not easy to use by inexperienced users.

The Context Metatags module is based on functionality from the Context module and allows setting meta tags for each individual Drupal page. It supports the Token module, so it is possible to build meta tags dynamically. It also provides some useful filters such as trimming on meta tag values.

2.2.3 Desired features for a module used by inexperienced users

In the case of a news outlet, non-end users usually assume the role of a blogger, author or editor. Considering that they have little experience with Drupal, some desired characteristics of a module to support their workflow would be:

- No App ID should be required to set the module up, but it should be an option so that integration with the various analytics platforms can be achieved.
• Single point of entry for editing, adding content and tagging. Single configuration pages are useful but they come with an overhead, both when it comes to workflow efficiency, and learning.
• No manual modifications on template.php or page.tpl.php, node.tpl.php etc. to add namespaces to the HTML header, a tweak required by some of the aforementioned modules.

2.2.4 The future: Drupal 8

Drupal 8 will support RDFa markup that uses terms from the Schema.org vocabulary. However there are still some remaining problems in Drupal 7’s core RDF that need to be fixed (Clark, 2013):
1. It is not possible to represent the data of compound fields, such as the address field, but only data of simple fields like Number.
2. When using layout tools like Views, Panels, etc. the data usually gets disarranged so that the fields that are supposed to be properties of the node are instead represented as properties in the view.
3. Themers can easily corrupt a site’s data when altering the way a field is displayed. In this line of thought, JSON-LD instead of RDFa has been proposed as an option that would remove the complexity from the field formatters and the theme layer. Additionally, no RDFa parsers would be required and lower budget consumers would be able to use the data since almost all modern platforms and languages handle JSON out of the box.

2.3 Conclusion

This chapter has outlined the concept of the Semantic Web and the building blocks that underpin it. Since Drupal is the platform where Semantic Enrichment will be performed, core and contributed modules of interest are recounted, keeping the nature of the target user group always in mind. Among these, Metatag displays the most potential. It becomes obvious that the field is in a state of constant change because new commercial technologies arise, standards get deprecated and because Drupal itself is undergoing major changes. Page level enrichment is chosen over individual level enrichment because it entails less manual labor, it is less ambiguous, and it is the only choice if commercial vocabularies are to be used.
3. Methods and Methodology

This chapter outlines the methods that were employed to collect data from the users, as well as the means for systematically analysing this data corpus. A user-centered approach is adopted since user input during the construction of an information system is a well-accepted necessity (Lazar, 2001, p. 5).

3.1 Artifacts employed in User Centered Design

3.1.1 Prototypes

Prototypes serve a variety of purposes such as assessing the technical feasibility of an idea, clarifying vague requirements, conducting usability testing and evaluation, underpinning discussions with stakeholders, exploring alternatives and confirming that a certain design direction is right (Preece, Sharp, & Rogers, 2002, p. 241). Depending on the stage of the design and development process different types of prototypes are appropriate.

3.1.1.1 Low fidelity prototypes

Low fidelity prototypes do not resemble the final product in a large extent, but are useful as they are simple, inexpensive and quick to modify so that alternative designs and ideas can be explored. Moreover, low fidelity prototypes elicit feedback that focuses on high level concepts rather than execution and stylistic choices. Prototyping in early stages of development ensures that the user will not discard the final product. Low fidelity prototypes are often paper-based and may range from a series of hand-drawn mock-ups to printouts, to wireframes.12 Nielsen (2003) recommends using paper prototypes as early as possible as it is cheaper to make changes before any code has been written than waiting until the implementation is complete. However, low fidelity prototypes pose navigational and flow limitations, have limited utility after requirements are established and are facilitator driven (Preece et al., 2002, p. 246).

3.1.1.2 High fidelity prototypes

High fidelity prototyping uses materials similar to the final product. High fidelity prototypes usually are computer-based, offering almost complete functionality. They clearly define a navigational scheme, can be used for testing and collecting true human performance data, are user driven and serve as a living specification. It is however not effective for requirements

User centred design calls for a collaborative design process where users deliver feedback regarding the prototypes and the final product. In the context of this thesis, the users were called to evaluate a high fidelity prototype in several rounds. Questions and themes that were used in the evaluation revolved around the perceived benefits of the product, positive and negative reactions towards interface elements, awareness, recall and intention of use.

3.2 Techniques for data gathering

3.2.1 Interviewing

Interviewing has been a primary means through which researchers gather qualitative data in order to understand user needs and behaviors and evaluate situated use of technology. Crang and Cook (2007, p. 60) describe the various levels of structure that an interview may have. In a highly structured interview all questions are predetermined and an order is specified, although a variety of answers may be expected. An unstructured interview is more akin to a conversation with a particular purpose but little focus. Semi-structured interviews fall between these poles, since the subject matter will be planned ahead of time, but lines of enquiry will be pursued within the interview to follow interesting themes that emerge. Having a preset guide means that the qualitative data from all the interviews will be more reliable, comparable, and easier to analyze in unison. Moreover, the interviewer has a level of control and a feeling of being competent, while being able to maintain an informal and more natural tone of discussion that allows following different topical trajectories.

In open-ended questions the informants give free-form answers, thus creating richer, but harder to analyze data in comparison to close-ended interviews. The most significant benefit of open-ended interviews is finding more than what was anticipated, as the informants might reveal mental models, problem-solving strategies etc. (Farrell, 2016). Open-Ended questions are appropriate when conducting design research, for exploratory studies and for qualitative usability testing (ibid.).
3.2.1.1 Open-ended, semi-structured interviews

This work employed semi-structured, open-ended interviews in the exploratory phase of the design lifecycle. The aim was determining the goal of the website, the experience level of the users, their workflow and plans for the future of the website so that an appropriate module could be chosen for customization. The verbal data was recorded, transcribed and discursively analyzed. Notes were jotted down, but only sporadically, so that rapport could be built with the informants.

Interviews as a research tool are however time consuming and it may not be feasible to talk with all the members of a group. Moreover, they tend to elicit only one person’s perspective, so universality of the informants’ comments is not ascertained (Shneiderman, Plaisant, Cohen, & Jacobs, 2013, p. 154). Finally, a drawback can be that interviews tend to give insights into what informants say they do, which might not be the same with what they actually do (Kellingley, 2016).

3.2.1.2 Alternatives to interviews

As an alternative to interviews, it can be revealing to gather a focus group of stakeholders to discuss issues and requirements (Preece et al., 2002, p. 213). The method assumes that individuals develop opinions within a social context by talking with others. A preset agenda is developed to guide the discussion and a facilitator to follows unanticipated issues as they are raised, prompt and directs discussion. Nielsen (1997) suggests that “in system development, the proper role of focus groups is not to assess interaction styles or design usability, but to discover what users wants from the system”.

Furthermore Loviglio (2012) advises against focus groups during a design iteration. Another disadvantage is that biased results may be yielded due to group influences. Since development was based on an existing module, which after alterations and additions served as a high fidelity prototype, it was not necessary to employ focus groups.

3.2.3 Thinking Aloud Protocols

An effective technique in order to gain insight into the participants’ thought process during usability testing is to invite representative users to think aloud about what they are doing as they are performing representative tasks (Shneiderman et al., 2013, p. 141).

During the concurrent thinking-aloud sessions, the practitioner should retreat into the background and only be supportive of the participants, not taking
over or giving instructions, but prompting and listening for clues about how they are dealing with the interface. Besides intervening as little as possible, Boren and Ramey (2000) suggest that only “hard” verbal data should be collected and analyzed. Participant introspection, inference, or opinion should not be valued or actively elicited. The participant should be encouraged to speak constantly as if alone in the room, without regard for coherency, and prompts should come after a predetermined period of silence and should be as terse and nondirective as possible. After a suitable time period for accomplishing the task list has elapsed, the participants can be invited to make general comments or suggestions, or to respond to specific questions (Shneiderman et al., 2013, p. 142).

Nielsen (2012) advocates the use of thinking aloud protocols since the practitioner can uncover the users’ misconceptions “which usually turns into actionable design recommendations”. The informal atmosphere allows for spontaneous suggestions for improvements. Furthermore, the technique is cheap, robust and flexible. Even if the practitioner does not follow the protocol with full precision, good results may still be gleaned. Additionally the technique may be used at any stage in the development lifecycle. Some disadvantages are that the situation may be unnatural and even though prompts and clarifications are necessary, such interruptions may alter user behaviour.

A similar technique is the retrospective thinking aloud protocol (Shneiderman et al., 2013, p. 143). After completing a task, users are asked what they were thinking as they performed it. The drawback is that the users may not be able to fully and accurately recall their thoughts, however this approach allows them to focus all their attention on the tasks they are performing and generates more accurate timings. Timing should be considered when using think aloud techniques. The standard think-aloud procedure may alter the true task time, as verbalizing the thought process adds extra cognitive load and the users may pause as they vocalize their thoughts.

In the context of this thesis, thinking aloud sessions tested the usability of the high fidelity prototype. Timing and notes were kept, while the verbal data produced were recorded and analysed with the proposed methods.

3.2.4 User Acceptance Testing

The goal of User Acceptance Testing is to validate that a system is of sufficient quality to be accepted by the users (Cimperman, 2006). Quality in
this case is the degree to which the system meets the stated and implied requirements in such a way to produce a usable system. User Acceptance testing does not include searching for every coding defect, and is not a discrete stage of the system’s lifecycle between development and testing. It is rather the idea that the user should be represented in every step of the software delivery lifecycle. By involving the users both in the design and system testing it is easier to uncover incorrect assumptions and inaccurate implementation of requirements early on. Various types of User Acceptance testing exist (ibid.), but in each case tests are not performed by members of the IT team, but a subject matter expert; like the owner of the solution. The criteria for passage are based on what makes sense in a real world environment rather than meeting rigid functional requirements:

- In basic UAT the future users of a system perform typical functions in a test environment and report issues.
- In a pilot test a subset of a future user group goes live with the new software to perform their normal functions. Their feedback is incorporated into the future version of the system.
- In friendly user tests, real users are selected to use the new software while being observed closely and having issues tracked.
- In beta tests real users are selected to use the new software and report back any issues or suggested enhancements.
- Parallel use and incremental rollout tests require performing work in both new and old versions of the system.

In the context of this thesis, User Acceptance Testing was performed with the Project Leader. A hybrid of basic and pilot UAT test was carried out, where the Project Leader was asked to perform tasks central to her daily workflow. Finding which exactly these tasks are was a result of applying thematic analysis in the data corpus from the open-ended, semi-structured interviews. Variance in action was allowed, and the feedback that could not be implemented in the remaining time, is listed as subject matter of future work.

3.2.5 Observations

Observing is useful at any time during product development. Early on in the process it helps designers understand the users’ needs, while other types of observation may examine whether the developing prototype meets those requirements. Observers can be outsiders or insiders in the field, and observation may be conducted in an overt or covert manner (Preece et al.,
Preece et al. (ibid.) stress that an observation should be driven by goals and questions, and suggest that practitioners employ a framework to structure and focus the observation. Such a framework pays attention to the space, the actors and activities, the artifacts, the events and the goals. They also recommend going through notes and records as soon as possible after each evaluation session to flesh out details and check for ambiguities. The practitioners should be aware of their personal opinion at all times and separate it from facts. Finally, they remark that data collection and analysis happen simultaneously, and the study might have to be refocused as the practitioners analyze and reflect on what they observe.

In a case such as this, it would be useful to conduct an overt non participant observation in the working environment of the users, with minimum disruptions to their routines. It could focus around the interaction of the users with the module to understand how and to which extend it is being used. This is however challenging, because when given the choice, all users preferred to conduct all the sessions involved in this work outside of their environment.

3.3 Experimental Design

The traditional scientific method consists of posing a question, conducting background research, constructing a hypothesis, testing the hypothesis by experimenting, analyzing data, drawing a conclusion and communicating the results. The power of the traditional scientific method can be employed in the study of interfaces (Shneiderman et al., 2013, p. 161). The term experimental design refers to a plan for assigning experimental units to different conditions\(^\text{13}\). It allows for making inferences about the relationship between independent variables and a dependent variable, it allows ruling out alternative explanations due to the confounding effects of variables other than the independent ones, and reduces variability within treatment conditions. When applied to human computer interaction, a tightly controlled environment is desired. The researcher deals with a practical problem and considers the theoretical framework, states a lucid testable hypothesis, identifies a small number of measurable dependent variables, judiciously chooses sample size, participants, and assigns them to groups, controls biasing factors, applies statistical methods to data analysis, resolves the problem and thus refines the theory and provides advice for future research.

In the context of this thesis, logging data, as well as A/B type tests where competing designs are compared, could underpin the experimental design. Statistics about how and to what extent the module is being used can be analyzed. Another area of interest is examining whether the claims regarding the benefits of Semantic Enrichment apply in this case. This is however subject matter of future work as this thesis assumes a qualitative stand, and the system has not been rolled out long enough to collect substantial amounts of data.

3.4 Ethics and Informed Consent

At each stage, participants were informed of the purpose of the study and of their right to withdraw at any time. They were provided with a written information sheet outlining the purpose of the study, what is expected of them, how their data will be stored and used, and how findings will be reported. Written informed consent was gathered and the records were kept securely, separate from data. Privacy and confidentiality were protected in the second round of usability testing since it was easy to extract the identities of the informants in the open-ended semi-structured interviews. Usernames were not collected and anonymity was otherwise preserved as much as possible. The forms of informed consent can be found in appendix C.

3.5 Analysis

The collected data exists in the form of field notes and audio files. According to Crang and Cook (2007, p. 132) data is partly made sense of and analysed through the focusing and refocusing of research aims and questions, the formal phrasing of the research to address specific issues, the methods used and the kind of data these yield, the individuals and groups chosen, as well as the issues and keywords jotted down during the sessions. The goal of the analysis however is to contextualize, assemble and order the data to “translate a messy process into a neat product” (Crang & Cook, 2007, p. 133).

The first step of the analytical process will be transforming the data to a form that is easier to work with. The field notes should for example be organized, and the audio files transcribed. Subsequently some form of annotating and coding will take place, and this can be done in different levels of detail. Units of data will be identified such as single words, phrases, extended utterances, and then the codes will be compared and contrasted to
construct an analytical narrative. Grounded Theory and Thematic Analysis employ various ways of coding data to frame the analysis.

3.5.1 Grounded Theory

Grounded theory is a research method that seeks to develop a theory that is grounded in data systematically gathered and analysed. According to Martin and Turner (1986), grounded theory is “an inductive, theory discovering methodology that allows the researcher to develop a theoretical account of the general features of a topic while simultaneously grounding the account in empirical observations or data.” The methodological stages\(^\text{14}\) of grounded theory are:

1. Identifying the area of interest.
2. Collecting data (through observations, accessing records, conversing with individuals or groups etc.)
3. Applying open coding on the data during collection.
4. Writing memos throughout the entire process. Theoretical memos should be written about codes and their relationships with other codes.
5. Conducting selective coding and theoretical sampling once the core categories and main concerns are recognized.
6. Sorting the memos and finding theoretical codes to organize the area codes.
7. Reading the literature and integrating it with the theory produced during the analysis.
8. Writing up the theory.

The major difference between grounded theory and other methods is its specific approach to theory development.

3.5.2 Thematic Analysis

Thematic analysis is a flexible, layered, iterative approach to deepen engagement with the qualitative data and discover themes (Braun & Clarke, 2006). Braun and Clarke (2006) recognize six phases of thematic analysis.

1. Familiarizing oneself with the data: the data should be transcribed, read and re-read, and initial ideas should be noted down.

\(^{14}\) [http://www.groundedtheoryonline.com/what-is-grounded-theory](http://www.groundedtheoryonline.com/what-is-grounded-theory)
2. Generating initial codes: interesting features of the data should be coded in a systematic fashion across the entire data set. Searching for themes: codes should be collated into potential themes, gathering all data relevant to each potential theme.

3. Reviewing themes: the themes should work in relation to the coded extracts and the entire dataset, and a thematic map of the analysis should be generated.

4. Defining and naming themes: the themes and the overall narrative should be iteratively refined.

5. Producing the report: will require a further level of reflection on the themes, the narrative and the examples used to illustrate themes.

In contrast to Grounded Theory, where the data collection and analysis are interleaved, thematic analysis assumes that a dataset already exists, and focuses attention on how that data might be analysed. Moreover thematic analysis implies that researchers need not subscribe to the implicit theoretical commitments of grounded theory if they do not wish or cannot develop a theory. Flexibility and simplicity is the main reason why thematic analysis has been chosen in the context of this thesis.

### 3.5.3 Coding

Coding, the process of naming or labeling entities, is an essential part both in Grounded Theory, and Thematic Analysis. It can be executed very systematically, but also more informally; like in this work. Different types of coding are described below:

- Open coding is the part of the analysis concerned with identifying, naming, categorizing and describing phenomena found in the text. It is focused on the text, and forms the basic units of the analysis by answering questions like “what is being referenced in this passage?” (Biddix, 2009).

- Axial coding is the process of relating codes to each other by asking questions regarding the conditions that caused or influenced the phenomenon, the context, and the effects or consequences (ibid.).

- Selective coding is the process of choosing one category to be the center around which all other categories are accumulated.

The analytical part of this work is constructed around open coding, yet elements of selective coding are also present, since some codes drive the design iterations.
3.6 Conclusion

This chapter has outlined the methods used to elicit the users’ needs, evaluate the design artifacts that lead to the final module and ensure that the final product can be taken in use. The chapter also proposes methods that can be applied in the future to assess the credibility of the claims about the benefits of Semantic Enrichment. The data yielded by each method is open coded and thematically analyzed to produce the findings in Chapter 6. The whole process is summarized in the figure below:

![Figure 3 Methods, Artifacts and Analysis](image-url)

Figure 3 Methods, Artifacts and Analysis
4. Design Guidelines

This chapter presents a typical life cycle model for user-centered web development, enumerates suggestions from the literature for designing forms – the main visual element of the module, and identifies guidelines for creating semantic authoring interfaces. Moreover, it outlines how Gamification strategies like scaffolding are utilized in order to further motivate and engage the users. The chapter also explains how the module conforms or deviates from the guidelines, and provides an argument for loosely following the rules.

4.1 User Interface Design for the Web

4.1.1 User-Centered web development

User-centered web development focuses on designing web resources that meet the functionality and usability needs of the end user. Functionality implies that a website must provide resources that users want, while usability means that a web site should be easy to use (Lazar, 2001, p. 5).

A typical life cycle model for user-centered web development includes the following stages (Lazar, 2001, p. 15):

1. Definition of the mission of the website, the user population, and the usability needs. Designing should begin with an understanding of the intended users, including population profiles that reflect their age, gender, physical and cognitive abilities, education, cultural or ethnic backgrounds, training, goals and personalities. Refinable classes to group the user population, besides their roles in the system, are novice or first time users, knowledgeable intermittent users, and expert frequent users (Shneiderman et al., 2013, p. 63).

   In this case, the mission of the website was delineated during the open-ended, semi-structured interviews, where all three informants agreed that it serves not only as a news outlet, but as a promoter for the faculty. The intended non-end users are editors, writers, and bloggers with a tertiary education, and while some of them are frequent users, they are not Drupal experts.

2. Collection of user requirements for the website (technological characteristics, browsers, speeds, content and information, qualities that will keep the users coming back or staying away, other preferences). Soliciting and clearly specifying the user requirements - the user community and the tasks they perform- is a major key to success in any development activity (Shneiderman et al., 2013, p. 101)
There was no direct elicitation of user requirements while developing the module since implementation started from a high fidelity prototype. However, preferences and firm requests from the users were collected during the thinking aloud sessions.

3. Conceptual and Physical design.
   In this work conceptual and physical design were intertwined as the high fidelity prototype was developed first, and then re-designed when feedback from the thinking aloud sessions was collected.

4. Execution of usability testing: Initially designers need to test and ensure that all functional and nonfunctional requirements are met. Subsequently, subjects from the target population evaluate whether the interface is easy to use, provide feedback, point problem areas and offer suggestions for improvements. Usability testing with the target population can be executed in usability labs, or by using techniques like thinking aloud and surveys. Usability experts may also be utilized to perform various types of expert reviews such as heuristic evaluations, guideline reviews, consistency inspections etc. Research recommends testing often and at varied times during the design cycle. Testing may be done at an exploratory stage when the designers try to conceive the correct design or as a validation effort to ensure that certain requirements are met (Shneiderman et al., 2013, p. 143).

   Usability testing was, as mentioned, conducted in the form of thinking aloud sessions, during the late stages of developing the high fidelity prototype. The user sample was successful in giving feedback and suggestions, and highlighting problematic areas. However, no usability experts reviewed the module. The findings of the usability testing are presented in Chapter 6.

5. Implementation and marketing.

6. Evaluation and improvement: Acceptance testing should be performed to verify adherence to measurable and explicit criteria such as rate errors by users, retention of commands, speed of performance, or subjective user satisfaction (Shneiderman et al., 2013, p. 152). Periodic evaluations of the interface’s effectiveness should also be performed. Some tools for evaluation are individual interviews that allow pursuing specific issues of concern, or professionally led focus groups that may elicit patterns of use or hidden problems. Furthermore logging data may provide usage frequency data, which in turn allows understanding why users are not using specific features, and guide maintenance and performance optimization (Shneiderman et al., 2013, p. 155).
Acceptance testing was performed with the head of the editorial team, while periodic evaluations that log quantitative data is an important part of the future work that can be carried out. Such data could for example draw attention to which meta tags are not being used, or which ones are used more often. In the first case the meta tags can be removed, while in the second stricter validation can be implemented.

4.1.2 Menu Selection, Form Fill In and Dialog Boxes

As this thesis intends to empower ordinary non-end Drupal users to perform Semantic Enrichment, there are certain constraints imposed by Drupal’s interface to the design. There is for example a pre-specified place in the editorial area of the website where the module may appear, and accordingly a standard area where notifications of status and error messages show up. These constraints however, come with the benefit of consistency, as all visual components like menus, forms and form elements, as well as status updates appear uniformly. Consistency, a principle described in paragraph 4.2.1 The eight golden rules, eases the learning curve and reduces cognitive load.

A few design suggestions by Shneiderman et al. (2013, p. 242) include to meaningfully group and sequence menu items, to carefully edit titles and labels, as well as laying these entities out appropriately to ease learning and increase selection speed. It is advised to logically group similar items, and cover all possibilities while using familiar terminology and clear distinctions. They (Shneiderman et al., 2013) further recommend presenting items in their natural sequence. Moreover, broad and shallow over deep and narrow structures should be preferred, brief and memorable keywords should be used, and shortcuts and typing ahead should be allowed. When filling in forms, Shneiderman et al. (2013, p. 247) propose that titles, labels and instructions are meaningful, comprehensible, familiar and consistent. The fields should be grouped and sequenced logically with proper alignment spaces. The required fields should be clearly marked. Cursor movement should be facilitated by the tab and arrow keys. Furthermore error prevention should stop the user from typing illegal inputs, and when an error for unacceptable values occurs, it should indicate the permissible alternatives. Finally, immediate feedback and completion signals should be implemented.

In this work, fields are grouped together by meta tag category, even though grouping by expected input type such as free text or URL has been discussed with the users. Especially after usability testing, titles, labels and instructions were re-written so that they are more understandable, coherent and draw
attention to the expected input. Cursor movement and completion signals are implemented by Drupal’s core. Implementing validators was one of the main augmentations to the original Metatag module, as well as indicating which fields are required. Stricter validation that goes past regular expression controls into validating that a URL is an active Facebook profile or Twitter account could be the subject of future work.

4.2 Design Guidelines

4.2.1 The eight golden rules

The eight golden rules of interface design as expressed by Shneiderman et al. (2013, pp. 70 - 71) are best practices derived by empirical studies. These guidelines advocate striving for consistency, catering to universal usability, offering informative feedback, designing dialogues to yield closure, preventing errors and offering simple constructive and specific instructions for recovery, permitting easy reversal of actions, supporting internal locus of control and reducing short term memory load. In this work attention was aimed at:

1. Keeping field labels and descriptions short and coherent. For example all fields expecting an image URL as input are labeled as: URL of the Image. The keyword placement is meant to draw attention to the expected input, a suggestion that came up during usability testing.

2. Prepopulating fields to avoid erroneous input and reduce memory load that would be required if the users had to copy and paste.

3. Implementing validation that indicates why the input caused an error, and highlighting the field where the error occurred.

4.2.2 Guidelines for Semantic Authoring

Khalili and Auer (2013) offer several design guidelines for Semantic Authoring tools. They define Semantic Authoring as the “tool-supported manual composition process aiming at the creation of semantic documents, or documents based on a non-semantic representation form, that are enriched with semantic representations during the authoring process”. They provide a set of quality attributes for designing suitable from the point of the consumer, user interfaces for semantic content authoring.

- Usability is defined as the effectiveness, efficiency and satisfaction with which users achieve goals. Factors such as learnability and utility are often added to the definition. Khalili and Auer (2013) state that simplicity is a prerequisite for usability, and thus, semantic authoring
tools should hide technical concepts from non-experts. In order to improve usability, a UI should offer a Single Point of Entry, Faceted Browsing, and Inline/View editing. A Single Point of Entry means that the environment where users annotate is integrated with the environment where they create, read, share and edit documents. This ensures that there is no added effort involved. Faceted browsing allows exploring information by applying filters. Finally, Inline/View editing allows users to edit items one by one, or in batches. Minimizing the required steps increases efficiency and user satisfaction. In this module, simplicity, learnability, and a single point of entry were applicable, and implemented.

- Personalization is the ability of a system to match the user’s needs and constrictions. Alani, Kalfoglou, O’Hara, and Shadbolt (2005) stress that users tend to be loyal to products they can customize to their preferences.
- Access control and support of standard formats are prerequisites for establishing collaborative practices (Khalili & Auer, 2013). The module enables fine grained access control, as permission for adding each meta tag can be added to every user role.
- Automation and Proactivity will reduce the need for human work and the possibility that users provide incomplete or erroneous metadata. A Semantic Authoring tool could offer prefilled forms, suggestions, default values and real time validation. Form prepopulation has been one of the most important enhancements to the original module. Suggestions in the form of tooltips were implemented in the high fidelity prototype, but deemed confusing by the users. Default values in the original module came in the form of token notation, which I assumed would be unfamiliar for the users, and were thus replaced with the actual value, for example instead of having [node:author:url] as a default value, the actual author URL can was extracted:
  
global $user;
  $uid = $user->uid;
  $user_url = $GLOBALS['base_url'] . '/user/' . $uid;

4.2.3 Loosely following the guidelines

The aforementioned guidelines will only provide an initial direction. It is more important to base the design upon a thorough understanding of the users’ tasks and work environments (Grudin, 1989). Grudin (1989) for example, argues against strictly following and prioritizing the consistency principle because a precise definition is composite and “elusive”, and “a fully consistent system is not achievable”. Since the target group consists of
inexperienced users, ease of learning will be a primary goal as explained in
the section to follow. But “ease of learning can conflict with subsequent ease
of use” (Grudin, 1989). Thus, it is ideal to provide enough flexibility to
permit a looser following of the guidelines as the users’ task knowledge and
experience increases. As a User-Centered design approach is followed, the
users’ perspective, feedback, and mental models will be prioritized above any
guideline.

4.3 Sustaining engagement with Gamification elements

Gamification refers to the application of game-based mechanics, aesthetics,
and design principles to non-game environments in order to engage people,
motivate action, promote learning, and solve problems (Kapp, 2012).

Designers adopting the user-centered design philosophy pay attention to the
user’s goals and strive to build products that help the user achieve these
goals in an efficient, effective and satisfactory manner. While these are
worthy goals, Gamification extends them by adding engagement,
empowerment, and showing the path to mastery (Kumar, 2013). The success
of the designer’s Gamification efforts depends on a clear identification and
understanding of the users, their motivation, and mission. Kumar (2013)
differentiates between intrinsic and extrinsic motivation. Intrinsic motivation
refers to internal motivation such as autonomy, mastery, and meaning. The
person finds the task or action rewarding in itself. Extrinsic motivation on
the other hand refers to external motivational techniques such as orders, job
duties, payment, trophies etc.

As mentioned in section 1.2.3 the level of motivation varies between the
different user types. During the open-ended, semi-structured interviews some
users disclosed that their main motivation for using the module is “rating
higher in the search engine results page”, a commercial and thus extrinsic
incentive. The design intends to promote the growth of intrinsic motivation
by employing the techniques described in the following section.

4.3.1 Onboarding, Scaffolding and Feedback

Onboarding and Scaffolding are two common processes within Gamification.
Onboarding refers to the act of bringing a novice into the system
(Zichermann & Cunningham, 2011, p. 59). When challenges are substantially
higher than a user’s skills and a steep learning curve is involved, anxiety
increases and this may demotivate user engagement (Nah, Telaprolu,
Rallapalli, & Venkata, 2013). Key goals of onboarding are to reveal the
complexity of the system slowly, to reinforce the user positively, and to remove opportunities to fail. While in many cases offering help or a brief introduction to the features and functions would be a valid onboarding tactic, Zichermann and Cunningham (2011, p. 60) suggest to “train and engage, but not overwhelm” the user. The system should not offer too much information, but rather facilitate an interactive first experience, since users do not read manuals and are likely to “skim the instructions” (Zichermann & Cunningham, 2011). These propositions align well with the nature of the target users and their varying level of motivation and dedication. The findings from the open-ended semi-structured interviews show however that some of the users consult often the help manual that exists on the site.

Scaffolding is meant to help the on-boarded yet inexperienced user advance to an expert stage. The amount of support should decrease proportionally to the increasing development of the user’s capabilities (Thomas & Berkling, 2013). The system should progressively disclose more features as the user gains more experience, cater for error prevention, and create a sense of accomplishment. Breaking a long arc of action into smaller, more achievable units may also be desirable.

Scaffolding is implemented through a status message that appears in Drupal’s notification interface that alerts the users once they have created a certain amount of nodes, that they now have unlocked a new meta data set. Hiding complexity is implemented in two ways. All the meta tag categories are in the beginning collapsed, a feature that the users approved of during usability testing. Moreover, depending on the Twitter Card type, relevant fields are shown or hidden.

Feedback is another important system design element. Broadly defined, feedback intends to inform users about where they are at the present time, ideally against a continuum of progress (Zichermann & Cunningham, 2011, p. 77). Provision of immediate feedback and making failure impossible at the first segments of actions can be considered as an onboarding technique. Feedback serves as a form of reinforcement as it shows progress, thus keeping the users interested and engaged (Nah et al., 2013). Feedback is only taken care of in the form of validation and success messages once content is created or updated, but multi-step form submission could also be considered for A/B testing.
4.4 Conclusion

The main purpose of this chapter was to delineate a set of guidelines for designing form based and semantic authoring user interfaces, as well as incorporating Gamification elements in the design to keep users motivated and engaged. The final design complies with these guidelines in the following ways:

1. Grouping meta tags by category.
2. The titles, labels, and descriptions were re-written to be simpler, more consistent and more familiar to the users according to the findings of the usability testing sessions.
3. Validation of user input which indicates why the input is illegal and highlights the field where the error occurred.
4. Indicating which fields are required.
5. Prepopulating fields to avoid erroneous input and reduce memory load.
6. Single point of entry: The environment where users add meta tags is integrated with the environment where they create and edit content.
7. Hiding complexity by keeping all categories collapsed when the module is displayed, and showing only the relevant fields based on the value of a select list.
8. Enabling fine grained access control where permissions for individual meta tags are given to each user type.
9. Implementing scaffolding by displaying a status message that alerts the users about ‘unlocking’ a new meta tag category once they have created a certain amount of nodes.
5. Implementation

This chapter describes the phases of the implementation process that started from the high fidelity prototype and led to the redesigned module. First, the issues that render Metatag unfit for the needs of Titan are outlined. Rectifying these issues lies at the core of development, and is one of the main contributions of this thesis. Subsequently the workflow, toolset and technologies are presented. The original module was forked, and the challenges that come with this decision are also analyzed. Finally, code snippets from the alterations and modifications are provided.

5.1 High Fidelity Prototype

The literature (Shneiderman et al., 2013, pp. 96-97) suggests an iterative design method that allows for early elicitation of user interface requirements, early testing of low-fidelity prototypes, revisions based on feedback from users and iterative refinements. However, as mentioned in Chapter 3, this work started with open-ended, semi-structured interviews, and continued with implementing a high fidelity prototype that underwent usability testing.

There are three reasons for not following the recommended design lifecycle. Firstly, Metatag already covers a part of the required functionality as well as having high adoption rates and being actively maintained and developed. Moreover, Drupal’s editing interface poses restrictions in the visual arrangement of the module, so, designing a low fidelity prototype with the users could have resulted in something infeasible. Last but not least, developing a module from scratch would be time consuming and challenging considering Drupal’s steep learning curve (McCourt, 2011, p. 34).

5.1.1 Rectifying Metatag’s flaws and inadequacies

Despite its popularity and variety of features, there are a number of issues that render Metatag unfit for the needs of Titan without modifications and additions. This thesis contributes with rectifying all of the following flaws and inadequacies:

1. Many deprecated tags are included both in the existing Metatag module, and the submodules. Examples of these are Photo Card¹⁵ and Gallery Card¹⁶ from the Twitter Cards set, as well as the Google Authorship

¹⁵ https://dev.twitter.com/cards/types/photo
¹⁶ https://dev.twitter.com/cards/types/gallery
metatags included in the Advanced Meta tags set. As stated by Mueller (2014) in his announcement of the cessation of the Authorship project, Google was seeing little difference in click behavior on search result pages with Authorship snippets compared to those without, and it “can even distract from those results”. Moreover, according to Enge (2014) since about half of Google’s traffic comes from mobile devices, it may not be wise to consume “valuable screen real estate for this type of markup on a mobile device”.

2. Many meta tags are irrelevant to the nature of Titan. This involves whole submodules but also subsets of meta tags from otherwise applicable submodules. Examples of these are the Twitter Card App meta tag set, or the Open graph Products submodule as a whole.

3. Many meta tags are appropriate only for webmasters, or are considered ‘difficult’ to fill in. Examples of these are the robots meta tags in the Advanced Metatag category, or the Dublin Core Advanced meta tag set.

4. There is a lack of explanatory elements like tooltips and help pages.

5. There is no validation of user input.

6. The token notation is used extensively, but may be too advanced for the target audience.

7. Some fields could have been prepopulated from standard Drupal fields.

8. There are no widgets or select elements.

9. The labels and descriptions contain typographical errors, grammatical mistakes, and can be hard to grasp.

10. There is a lack of indication of which fields are required.

11. The vertical-tab element displays a preview of all the meta tags added. This preview tends to be long and repetitive as the title, the description, the canonical URL, the image and its dimensions are some of the fields that appear in many meta tag sets.

12. The meta tag generation itself is incorrect. For example in the Dublin Core sub-module, elements that expect non-literal values are generated as meta tags, while they should have been link tags.

13. The interface of the module displays expanded elements, while testing has shown that users prefer these elements to be collapsed.

5.2 Workflow and toolset

The provisioned server; ifi-diw13.uio.no, was used for development, staging and setting up a Git repository for the project. Git is a tool helping to manage collaboration, configuration and code changes during the project life
cycle. Initially a Drupal sandbox project was created. Sandbox projects\(^\text{17}\) contain experimental code and have most of the features of full projects, such as a project page, a Git repository, an issue queue, as well as the ability to appoint co-maintainers. However the developer cannot create releases out of sandbox projects, and the only way to obtain them is via Git. The sandbox can be found in: [http://drupal.org/sandbox/magdalini/2699331](http://drupal.org/sandbox/magdalini/2699331).

Subsequently, the original Metatag module was cloned under:

```
/var/www/html/sites/all/modules with the command git clone -branch
magdalif https://git.drupal.org/project/metatag.git, where magdalif is the
name of the branch I was working on. Then development started in order to
implement the alterations and additional features described in chapter 5.4.
The files were accessed with FileZilla and edited locally, and then PuTTY
was used to perform the Git add, commit and push actions. Once the module
was ready for usability testing, it was compressed and copied with scp into
the test server which mirrors Titan.uio.no, where it was unpackaged and
enabled with drush.
```

![Workflow diagram](image)

**Figure 4 Workflow**

### 5.3 Technologies Used

#### 5.3.1 Languages

**5.3.1.1 PHP, Drupal's Form API**

Drupal is written in PHP, and all core code adheres to strict standards concerning indentation and whitespaces, structuring, naming and documentation. Following best practices increases the chances that the

\(^{17}\) [https://www.drupal.org/node/1011196](https://www.drupal.org/node/1011196)
The best practices I complied with can be found in Appendix D.

The parts of the initial module that I modified revolved around .inc and .module files for the core of the module and the sub-modules that were applicable to Titan’s case. I implemented some hooks in the .module files, but the additional functionality and alterations described in section 5.4.2 is found in the metatitan.metatitan.inc and metatitan.metatitan_submodule_name.inc files. In those .inc files I altered the existing implementations of hook_metatitan_info, which return an $info['tags'][metatag_name] with all the form elements. I also added validation functions, named _metatitan_submodule_element_validate, as well as helper functions called in the hook_metatitan_info implementations to provide the options of select lists. Finally, I implemented hook_form_alter() for some of the sub-modules.

As it becomes apparent, I mostly worked with the Form API, which simplifies handling forms since the developer creates an array and lets the engine generate the HTML. Since the form is represented as structured data, adding, deleting, reordering and changing forms is easy. Moreover, additional form validation or processing can be added to any form. A fuller account of the stages of initializing, theming, validating, submitting and dealing with the properties of a form can be found in Appendix D and may be useful to those looking to extend/maintain the module. An example of what a form element looks like follows:

```php
$info['tags']['og:image:type'] = array(
    'label' => t('Image type'),
    'description' => t('The format of the image'),
    'form' => array(
        '#type' => 'select',
        '#options' => _metatitan_opengraph_image_type_mime(),
        '#empty_option' => t('- None -'),
    ),
    'weight' => ++$weight,
) + $og_defaults;
```

### 5.3.1.2 jQuery

Using JavaScript adds dynamic presentation effects. In this case jQuery, a lightweight JavaScript library was used to prepopulate fields, display vertical sidebar information, and collapse elements. jQuery contains all the common DOM, event, effects and Ajax functions. Scripts are added using the drupal_add_js() function. Once again, several design patterns and best practices will be reviewed, maintained and adopted by the community. The

[18] https://www.drupal.org/node/171213
practices were followed. The ones that were consequential to the design of the code are enumerated below, but a fuller account can be found in Appendix D.

- DRY Principle\(^{19}\), single var pattern, using jQuery’s native ways to add and retrieve elements instead of JavaScript’s.
- Wrapping all code in a strict JavaScript closure, which means that undeclared variables will halt execution\(^{20}\):
  ```javascript
  /**
   * @file
   */
  (function () {
    "use strict";
    // All the JavaScript for this file.
  })();
  ```
- The code was organized by following the Module Pattern\(^{21}\). This design pattern has the advantages of privacy, because functions create scope. It allows for maintaining the private variables’ values, keeping only the module – function name available and leaking nothing else in the global scope, as well as promoting maintainability and reuse. All the functionality is contained in an IIFE (immediately invoked function expression) which returns an object accessed inside Drupal.Behaviors.
  ```javascript
  var myJavaScriptModule = (function(){
    var myJS = {};
    myJS.moduleMethod = function () {
      // implement functionality
    }
    return myJS;
  })();
  Drupal.behaviors.myModule = {
    attach: function (context, settings) {
      // Use the JavaScript Module.
      myJavaScriptModule.moduleMethod();
    }
  };
  ```
- DOM access is the most common performance bottleneck (Stefanov, 2010, p. 183). Accessing the DOM in loops was avoided, DOM references were assigned to local variables, and the length was cached when iterating over HTML collections. Doing so is anywhere between two and 190 times faster (Stefanov, 2010).
  ```javascript
  for (var i = 0, max = array.length; i < max; i++) {
    // do something with array[i]
  }
  ```

\(^{19}\) [https://learn.jquery.com/code-organization/dont-repeat-yourself/](https://learn.jquery.com/code-organization/dont-repeat-yourself/)

\(^{20}\) [https://www.drupal.org/node/172169](https://www.drupal.org/node/172169)

\(^{21}\) [http://atendesigngroup.com/blog/requirejs-and-javascript-architecture-drupal](http://atendesigngroup.com/blog/requirejs-and-javascript-architecture-drupal)
5.3.2 Dependencies

The final module depends on CTools and the Token module, while Potx was used to translate the interface. Token is used to develop custom tokens not supported by core, as well as providing a token browser window where the users could insert tokens into the field forms. However, usability testing has shown that this was too advanced a feature, so it was omitted in the redesign. Regarding the translation, after Potx was installed, a template file was generated in the “Extract” tab on the “Translate interface” administration interface. The desired strings were translated, and then the template file was uploaded back through the “Import” tab, with the choice to replace existing strings. The template file, metatag.nb.po is also uploaded in the module’s folder.

5.4 Alterations and additional features

5.4.1 Patching versus forking

The developed module is a fork of Metatag because the majority of the alterations and additions required were too specific to be useful for the Metatag community, even though the suggestions in section 5.1.1 are still general enough to be considered.

A fork is a copy of a repository. Forking allows the developer to start independent development and freely experiment with changes without affecting the original project. Most commonly, forks are used to either propose changes to someone else’s project or to use someone else’s project as a starting point for own ideas. Forking allows for customization but it splits the community, introduces duplication of effort and especially in this case, may produce incompatibilities (Robles & González-Barahona, 2012). Especially in the case of this module, refactoring in file, database, and variable names was essential for the customized module to run on the test server without collisions. This makes merging updates from the original module challenging, while manual work will be required upon Git pulls.

Patches are pieces of code that solve an existing issue. Patches describe the changes between a before and after state of a module (or core), and by applying the patch the issue should no longer exist. Patches maintain controllability over the project as they focus on a single change request and

---

22 https://www.drupal.org/patch
therefore are easily tested, reviewed and documented. Git commit access is not required to submit patches: a patch can be uploaded to an issue in the issues queue for anyone with a Drupal account. Typically the –dev versions are patched against. The project should be cloned, and then a branch should be created out of the local downloaded version. Once changes are made and committed, a patch_name.patch file can be created and then uploaded. As a best practice, an interdiff, a text file that describes the changes between two versions of a patch, should be created.

5.4.2 Phases of Development

The alterations and additions that rectified the defects mentioned in section 5.1.1 and resulted in the customized module were implemented during the following phases:

5.4.2.1 Singling out relevant and supported subsets and meta tags

Metatag comes with a core module that allows creating various tags relevant to search engines (robots, referrer, refresh, revisit, caching, and expiration), the deprecated Google Authorship tags and geolocation tags. Included in Metatag is an array of sub-modules that provide integration with other Drupal modules, or provide markup for commercial products, mobile applications etc. The deprecated and irrelevant to the nature of the website meta tags and sub-modules were discarded as they had no direct relationship the content described in each node, or were considered too advanced for the target users.

The table below lists the sub-modules and respective fields that were kept in the final implementation. The fields marked with gray implement validation, while the ones marked with light green have widgets. All fields that are eligible for Prepopulating are filled automatically. None of the aforementioned functionality was provided in the original module.
### Table 2 Sub-modules, fields, functions

<table>
<thead>
<tr>
<th>Basic meta tags</th>
<th>Google News prefix og:</th>
<th>Open Graph prefix og:</th>
<th>Twitter Cards prefix twitter:</th>
<th>Dublin Core prefix dcterms:</th>
<th>Google+ prefix itemprop:</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>news_keywords</td>
<td>type</td>
<td>card</td>
<td>title</td>
<td>itemprop:</td>
</tr>
<tr>
<td>description</td>
<td>standout</td>
<td>title</td>
<td>title</td>
<td>description</td>
<td>name</td>
</tr>
<tr>
<td>img_src</td>
<td></td>
<td>description</td>
<td>description</td>
<td>subject</td>
<td>description</td>
</tr>
<tr>
<td>canonical</td>
<td>article:author</td>
<td>site</td>
<td>publisher</td>
<td>headline</td>
<td></td>
</tr>
<tr>
<td>rights</td>
<td>article:publisher</td>
<td>creator</td>
<td>contributor</td>
<td>author</td>
<td></td>
</tr>
<tr>
<td>generator</td>
<td>article:section</td>
<td>url</td>
<td>creator</td>
<td>publisher</td>
<td></td>
</tr>
<tr>
<td></td>
<td>article:tag</td>
<td>img:url</td>
<td>source</td>
<td>image</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>see_also</td>
<td>img:width</td>
<td>relation</td>
<td>logo</td>
</tr>
<tr>
<td></td>
<td>url</td>
<td>img:height</td>
<td>type</td>
<td>datePublished</td>
<td></td>
</tr>
<tr>
<td></td>
<td>image:secure_url</td>
<td>img:alt</td>
<td>format</td>
<td>dateModified</td>
<td></td>
</tr>
<tr>
<td></td>
<td>image:type</td>
<td>player</td>
<td>language</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>image:width</td>
<td>player:width</td>
<td>identifier</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>image:height</td>
<td>player:height</td>
<td>rights</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>locale</td>
<td>player:stream</td>
<td>date</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>article:published_time</td>
<td>player:stream:content_type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>article:modified_time</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>profile:... book:...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 5.4.2.2 Finding and correcting implementations that do not conform to the standards

One of the phases of customizing the module involved finding out whether the outputted meta tags conformed to the standards. I found out that some Dublin Core tags should use `link` instead of `meta` elements. The official specification[^23] for expressing Dublin Core metadata using X/HTML describes the DC-HTML profile where each statement is represented as either a `meta` or `link` element. The choice depends on whether the statement contains a literal value surrogate, or a non-literal value surrogate. Literal value surrogates are made up of exactly one value string, either plain or typed. Non-literal value surrogates are composed of a value URI and zero or one plain value strings.

For example, DC’s user guide to publishing metadata\(^{24}\) defines the \texttt{dcterms:creator} property as one that has a range of non-literal values. A correct way to output this statement in a meta tag would be:

\begin{verbatim}
<link rel="DCTERMS.creator" href="http://example.org/agents/DCMI
title="Dublin Core Metadata Initiative" />
\end{verbatim}

However the output looks like:

\begin{verbatim}
<meta name="dcterms.creator" content="magdalini" />
\end{verbatim}

The issue \url{https://www.drupal.org/node/2710365} was created in the Metatag queue describing the problem. The response from one of the module’s creators was that the tags are outputted this way because of oversight and that they only allow one value, while some work is required to allow multiple values.

In order to fix this problem without having to spend a lot of time in developing the multiple value output, the OWMS profile described in paragraph 2.1.4.7 was used as it is a ratified governmental standard, provides examples, and recognizes many different output styles as legal. So, after the changes in the original module, the aforementioned output changed into:

\begin{verbatim}
<link rel="dcterms.creator" href="http://ifi-diw13.uio.no/user/2" />
\end{verbatim}

which conforms to the second of the three choices listed in OWMS:

\begin{verbatim}
<meta name="DCTERMS.creator" scheme="" content="" />
<link rel="DCTERMS.creator" href="" />
<link rel="DCTERMS.creator" href="" title="" />
\end{verbatim}

5.4.2.3 Iteratively changing titles, labels, descriptions, default values, adding widgets, translating

The labels and descriptions were simplified; default values, empty options and widgets were added, as well as indications of which fields are required. Once usability testing was executed more simplifications, regroupings and translations were applied iteratively. A translation in the .po file looks like:

\begin{verbatim}
#: metatitan.metatitan.inc:247
msgid "An image associated with this page, for use as a thumbnail in social networks and other services."
msgstr "Bildet som er tilknyttet innholdet. Brukes som miniatyrbilde i sosiale medier og andre tjenester. Feltet er forhåndsutfylt med URL-adressen til toppbilde."
\end{verbatim}

\(^{24}\) \url{http://wiki.dublincore.org/index.php/User_Guide/Publishing_Metadata#dcterms:creator}
Regarding regrouping meta tags, instead of the Basic and Advanced categories in the initial core module, the remaining tags were placed into the Basic and Google News categories. Screenshots from the final module can be found in Appendix E.

5.4.2.4 Adding validation functions

Validation functions were added for obligatory fields, and all fields accepting URLs, dates and twitter accounts. In the case of URLs, it is also checked if the URL corresponds to an image of acceptable type, a Facebook, or Google+ profile. Regular expressions were used, and not API calls that load the actual profile to see if it is valid or not. An example of a validation function follows:

```php
/**
 * Validate that the user input is an image URL.
 * @param $element
 * The form element to be validated
 * @param $form_state
 * An array that stores information about the form
 * @param $form
 * The form
 */
function _metatitan_opengraph_image_url_validate($element, $form_state, $form) {
    // if input exists, it must be a URL or token
    if (!empty($element['#value'])) {
        if (!preg_match('/\b(?:http|https|ftp):\/*|www\.*\[a-z0-9+@\/%?=~_|!:,.;]*\[a-z0-9+@\/%?=~_|]/i', $element['#value']) {
            $message = t($element['#value'].' requires you to enter a valid URL.);
            form_error($element,$message);
        } else {
            $format = substr($element['#value'], strrpos($element['#value'], '.') + 1);
            // MIME types
            if (($format !== "jpeg" && $format !== "jpg" && $format !== "gif" && $format !== "png")) {
                $message = t($element['#title'].' requires you to enter an image URL. Image URLs end in .jpeg, .jpg, .gif, or .png.');
                form_error($element,$message);
            }
        }
    }
}
5.4.2.5 jQuery: Prepopulating, Collapsing, Vertical Tabs

The original Metatag provides field prepopulation in the form of token notation as default values in fields. Tokens, e.g. \[\text{node:summary}\], automatically insert the corresponding information from that field or value. According to the findings of usability testing, token notation and token browsing was either confusing to the users, or went completely unnoticed. Moreover, token notation does not show dynamic changes on the field but values such as \[\text{current-page:url:absolute}\]. The figure below displays the token browser.

Available tokens

<table>
<thead>
<tr>
<th>NAME</th>
<th>TOKEN</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current date</td>
<td>Tokens related to the current date and time.</td>
<td></td>
</tr>
<tr>
<td>Current page</td>
<td>Tokens related to the current page request.</td>
<td></td>
</tr>
<tr>
<td>Current user</td>
<td>Tokens related to the currently logged in user.</td>
<td></td>
</tr>
<tr>
<td>Meta tags</td>
<td>Generated by the Metatag module, may not be used to fill in other meta tags.</td>
<td></td>
</tr>
<tr>
<td>Nodes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Author</td>
<td>[\text{node:author}]</td>
<td>The author of the node.</td>
</tr>
<tr>
<td>Authors</td>
<td>[\text{node-field_authors}]</td>
<td>Term reference field.</td>
</tr>
<tr>
<td>Body</td>
<td>[\text{node:body}]</td>
<td>The main body text of the node.</td>
</tr>
<tr>
<td>Caption</td>
<td>[\text{node-field_caption}]</td>
<td>Text field.</td>
</tr>
<tr>
<td>Category</td>
<td>[\text{node-field_category}]</td>
<td>Term reference field.</td>
</tr>
<tr>
<td>Comment count</td>
<td>[\text{node-comment-count}]</td>
<td>The number of comments posted on a node.</td>
</tr>
<tr>
<td>Content ID</td>
<td>[\text{node:id}]</td>
<td>The unique ID of the content item, or “node”.</td>
</tr>
<tr>
<td>Content type</td>
<td>[\text{node-content-type}]</td>
<td>The content type of the node.</td>
</tr>
<tr>
<td>Date changed</td>
<td>[\text{node:changed}]</td>
<td>The date the node was most recently updated.</td>
</tr>
<tr>
<td>Date created</td>
<td>[\text{node:created}]</td>
<td>The date the node was posted.</td>
</tr>
<tr>
<td>Edit URL</td>
<td>[\text{node-edit-url}]</td>
<td>The URL of the node’s edit page.</td>
</tr>
<tr>
<td>Feed Category</td>
<td>[\text{node-field_feed_category}]</td>
<td>Term reference field.</td>
</tr>
<tr>
<td>Guest blogger</td>
<td>[\text{node-field_guest_blogger}]</td>
<td>Term reference field.</td>
</tr>
<tr>
<td>Language</td>
<td>[\text{node:language}]</td>
<td>The language the node is written in.</td>
</tr>
<tr>
<td>Lead Paragraph</td>
<td>[\text{node-field_lead_paragraph}]</td>
<td>Long text field.</td>
</tr>
</tbody>
</table>

Figure 5 Token Browser

Prepopulating fields was one of the most common requests during the first round of usability testing and providing this functionality was a significant contribution of this thesis. A snippet of how the description related fields get prepopulated follows:

```javascript
var descContainer = getSummaryFields($);

// Retrieve the contents of the ckeeditor Sammendrag field
var ckData = '', ckInstances = CKEDITOR.instances['edit-body-und-0-summary'], editorType = $('\.text-summary.ckeditor-mod.form-textarea.text-summary-processed');

// Once the contents of the sammendrag field change
```
editorType.change('input', function () {

    // fill description related fields with new content
    ckfill($, ckData, ckInstances, descContainer);
});

/**
 * Get an array of the text area fields whose id contains
 * the word 'description'. These correspond to summary related
 * summary related metatags
 *
 * @return textareas of description related metatags
 */
function getSummaryFields($) {
    var descContainer = [];
    var textareas = $('textarea');

    for (var i = 0, len = textareas.length; i < len; i++) {
        if (textareas[i].id.contains('description') &&
            textareas[i].id.contains('metatitans')) {
            descContainer.push(textareas[i]);
        }
    }

    return descContainer;
}

/**
 * Get the text in the ckeditor and fill in the collection
 * defined by container
 *
 * @param ckData The data in the instance of CKeditor
 * @param ckInstances The instance of the editor (there are 2)
 * @param container The metatag fields to be prepopulated
 */
function ckfill($, ckData, ckInstances, container) {
    // if ckeditor is activated
    if (typeof(ckInstances) !== 'undefined') {
        if (ckInstances.document !== 'undefined') {
            ckData = ckInstances.getData();
        }
    }

    // stripping html tags from entries
    div = $('<div></div>');
    div.innerHTML = ckData.replace(/(<([^>]+)>)/ig, '') || '';

    // change the value of all description fields.
    prepopulateCollection($, div.innerHTML, container);
}

The table below demonstrates how standard Drupal fields are used to
prepopulate the meta tag fields, and which function performs the respective
action (helper functions not included). Changes at any point in the standard Drupal fields fire events so that the meta tags are always updated.

Table 3 Prepopulating the meta tag fields

<table>
<thead>
<tr>
<th>META TAG FIELDS</th>
<th>STANDARD FIELDS</th>
<th>ID/CLASS/CODE</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sidetittel</td>
<td>Tittel</td>
<td>#edit-title</td>
<td>prepopulateTitles</td>
</tr>
<tr>
<td>Beskrivelse</td>
<td>Sammendrag</td>
<td>CKEDITOR</td>
<td>prepopulateSummaries</td>
</tr>
<tr>
<td>URL-adressen til bildet Bildets bredde Bildets høyde Bildetype Alternativ bildetekst</td>
<td>Toppbilde from the Scald dropzone</td>
<td>.image-widget-data</td>
<td>prepopulateImg</td>
</tr>
<tr>
<td>Forfattere</td>
<td>Current user OR Forfatter</td>
<td>#edit-name OR #edit-scald-authors-und</td>
<td>prepopulateAuthor</td>
</tr>
<tr>
<td>Overskrift</td>
<td>Ingress</td>
<td>#edit-field-ingress-und-0-value</td>
<td>prepopulateHeadlines</td>
</tr>
<tr>
<td>Tagger</td>
<td>Tagger</td>
<td>#edit-field-tags-und</td>
<td>prepopulateTags</td>
</tr>
<tr>
<td>Publikasjonsdato, Modifikasjonsdato</td>
<td>Tidsplanlegging svalg, Formfatterinfor masjon</td>
<td>#edit-date, #edit-publish-on, the current date</td>
<td>prepopulateDates</td>
</tr>
<tr>
<td>Kategori, Tema</td>
<td>Kategori</td>
<td>Contains #edit-field-categories-sn-OR #edit-field-subsite-tag-</td>
<td>prepopulateCategory</td>
</tr>
<tr>
<td>URL-adressen til skaperen</td>
<td>-</td>
<td>$GLOBALS['base_url'] .'/user/'. $uid;</td>
<td></td>
</tr>
<tr>
<td>URL-adressen til innholdet</td>
<td></td>
<td>location.href.substr(0,location.href.lastIndexOf(&quot;/&quot;));</td>
<td>prepopulateUrls</td>
</tr>
<tr>
<td>Språkform</td>
<td>-</td>
<td>nb_NO</td>
<td>prepopulateLocale</td>
</tr>
<tr>
<td>Type, Kategori, Innholdstype,</td>
<td>-</td>
<td>#article-node-form #blog-node-form #shortarticle-node-form</td>
<td>prepopulateType</td>
</tr>
<tr>
<td>URL-adressen til medieavspilling</td>
<td>Video link</td>
<td>#edit-field-videolink-und-0-url</td>
<td>prepopulateVideo</td>
</tr>
<tr>
<td>URL-adressen til Utgiveren</td>
<td>Static: <a href="http://mn.uio.no/">http://mn.uio.no/</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>URL-adressen til utgiverens Facebook profil</td>
<td>Static: <a href="https://www.facebook.com/Titanuiono">https://www.facebook.com/Titanuiono</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nettstedets Twitter-konto</td>
<td>@unioslo_titan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rettigheter</td>
<td>Static: Alle rettigheter forbeholdt</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

63
In addition to field prepopulation, a JavaScript file was developed to keep all the sub-modules collapsed, another request from the users. Finally code to modify the previews of the added meta tags in the vertical tabs is also provided but disabled in conformance to the findings of usability testing.

### 5.4.2.6 Gamification elements

The meta tags in the Google News subset can be ‘abused’ if the users insert non appropriate values, or if they fill in the standout meta tag more than seven times per calendar week. During usability testing a lot of clarifications were required for this category, so I followed the approach described in section 4.3.1, where I let the Google News category be visible only for ‘trusted’ users, where a user is considered trusted if they have created n amount of articles, blogs or smånytt. Selecting the amount of nodes is arbitrary now, and can change once quantitative data is gathered. In the same line, if quantitative data proves that some meta tags are not used properly, these can be added in the ‘unlockables’.

```php
/** *
 * Returns the number of nodes created by the user *
 */
function _metatitan_nodes_created_by_user ($uid) {
    // if this is an edit page
    if (arg(2) == 'edit') {
        // make sure that these nids correspond to articles, blogs, smånytt
        $query = new EntityFieldQuery();
        $query
            ->entityCondition('entity_type', 'node')
            ->entityCondition('bundle', array('article', 'shortarticle', 'blog'))
            ->propertyCondition('uid', $uid);

        $result = $query->execute();
        if (isset($result) && count($result) > 0) {
            $nids = array_keys($result['node']);
            if (isset($nids)) {
                return count($nids);
            }
        }
    }
    return -1;
}
```

Alternatively the number of meta tags created could have been returned, but the approach was avoided because `node_load`, `node_load_multiple` and `entity_load` took a long time to execute:

```php
if (isset($nids)) {
```
$user_nodes = node_load_multiple($nids);
$tags = array();
foreach ($user_nodes as $ncreated) {
    if (property_exists($ncreated, 'metatitans')) {
        $t_created = array_values($ncreated->metatitans)[0];
        $tags[] = count($t_created);
    }
}

if (isset($tags)) {
    for ($i=0; $i<sizeof($tags); $i++) {
        if ($tags[$i] >= $threshold) {
            return $tags[$i];
        }
    }
}

Once the threshold has been exceeded the user is presented with a message in the standard Drupal status bar:

Figure 6 Unlocking a set of meta tags

5.4.2.7 Advanced permissions

Providing fine grained permissions for each individual meta tag was a matter of discussion both with the Product Manager and the Project leader. This functionality offers flexibility, but makes the permission assignment process laborious. The following figures display how to enable the extended permission feature. The feature is available though the administrative interface.
5.4.2.8 Content types available for meta-tagging

It is possible to select which content types will be enriched since it is not valuable to annotate nodes aimed at internal use or nodes that need not appear in the SERPs and social media feeds. Thus, the content types chosen for enrichment are articles, blogs and short articles (smånytt). The factoids (Visste du at) display only one line of content so there is no benefit in enriching them. The feature is available through the SEO part of the administrative interface.
5.4.2.9 Help material

Since the users rejected the tooltips and showed preference to the help pages (brukerhåndvisning) I have created a help page that clarifies the scope of the module and explains where the interface can be found, and how to use it.

5.5 Output

Once an available content type has been enriched and saved, meta tags are outputted in the <head> of the page. A code snippet of such an output follows:

```xml
<meta name="description" content="Sand og vann kan være forbløffende komplisert. En ny studie viser hvordan propper dannes i rør. Denne kunnskapen er viktig for oljeutvinning, spredning av forurensing og transport av blod i kroppen.” />

<link rel="canonical" href="https://titan-app-test01.uio.no/node/1768" />
<meta name="generator" content="Drupal 7 (http://drupal.org)” />
<meta property="og:title" content="Ny kunnskap om propper i rør viktig for industrien" />
<meta property="og:site_name" content="Titan” />

<meta property="og:type" content="article" />
<meta property="article:publisher" content="https://www.facebook.com/Titanuiono-836558236423265" />
<meta property="article:section" content="Naturvitenskap” />
<meta property="article:tag" content="Fysikk” />
```
It is also valuable to validate the page using the following tools, which accept full URLs and not code snippets.

1. https://cards-dev.twitter.com/validator

5.6 Conclusion

This chapter outlined the implementation of the final module. The recommended iterative design lifecycle that calls for early user involvement and low fidelity prototyping was not followed since Metatag covered a lot of the desired functionality and there were time limitations. However many modifications and additions were required since several tags were deprecated, irrelevant or too advanced. Moreover there was a lack of interface elements that enhance the user experience and last but not least the meta tag generation itself was not conforming to the standards.

Development-wise, Git was used to configure and manage the code changes in the Drupal sandbox project which was later refactored and launched on Titan’s test server. Drupal’s Form API and jQuery were the two main
technologies used. The development happened in phases, some of which were iterative redesigns based on the findings of usability testing and user acceptance testing. All in all, the appropriate meta tag subsets were singled out and made compliant to the standards, the fields and their descriptions were re-written, re-grouped and translated, default values, widgets, validation functions and field prepopulation was added, as well as Gamification elements and fine grained permissions.

Some reflections from the process involve using an existing module as a high level prototype. Adhering to the timeframes was one of the reasons for choosing to follow this approach, but I found that when getting involved with an open source code base, it can be challenging to have an overview of what already is there, and why some performance problems occur. Moreover, being based on an existing module limits the design choices. Finally, choosing to fork over patching will be a challenge for the future.
6. Findings

This chapter outlines how the methods and methodologies generated and analyzed the qualitative data corpus. The objective was to discern the nature and goals of the website, the workflow, mental models and expectations of the target user group, as well as to uncover ambiguities and misconceptions around the interface, and to get feedback. The findings provided guidance for the iterative re-design of the high fidelity prototype, as well as answers to the research questions. The appropriateness and sufficiency of the methods and methodologies is assessed in the end of the chapter.

6.1 Open-Ended, Semi-Structured Interviews

As mentioned in section 3.2.1, open-ended, semi-structured interviews were chosen as the tool of inquiry during the exploratory phase of the module’s design. The purpose was to understand the goal of the website, the mental models and experience level of the users, their workflow, and the future plans for the website. The full transcripts can be found in Appendix A.

6.1.1 Sample, Interview Guide and Context

Due to the nature of the questions for this interview round, I considered gathering a representative sample of users impractical. Interviews were thus held with the main stakeholders; the Product Manager, the Project Leader, and the Communication Responsible. The interview guide was structured around the following range of topics:

- The goal of the website.
- The project’s timeline and plans for evolution
- The informant’s background, motivation, role in the project and daily workflow.
- Technical questions. In the interview with the Product Manager, I wanted to understand what kind of functionality is relevant and can be supported, if there are any prerequisites and rules I should follow, and how I could get started with development. In the cases of the Project Leader and the Communication responsible, who I consider to be the most advanced users, I tried to probe their experience level. I presupposed that since they are the most advanced, their experience level will set a threshold on how detailed enrichment should be.
- Preferences regarding design matters, such as having limited choices, and manuals.
- Difficulties and requests for changes.
I kept a conversational style and a few more themes than I planned emerged. I tried not to include leading questions, ask ‘how’ and ‘what’ questions, and keep the interviews shorter than an hour. The informants were encouraged to choose the location and time that is most convenient to them. I held one of the sessions in the Product Manager’s office, and the rest in colloquium rooms in the Department of Informatics, where no other listener was present.

6.1.2 Thematic Analysis

As instructed by the Thematic Analysis method, I first familiarized myself with the data by re-writing the notes and transcribing the records right after each session. I also wrote a short reflection in order to retain all the data and impressions that emerged during the interaction. I then highlighted prominent words and assigned an open code in charged phrases. The open-codes were subsequently grouped into candidate themes. The candidate themes were refined, merged or emitted depending on whether there was enough data to support them. My intention was to create internal coherence in a theme, but strong distinctions between themes. Afterwards the themes were organized in the thematic table below. The themes occupy a row each, by order of appearance. Keyword categories correspond to each theme, and the charged phrases are presented together with the number of times the keyword category was mentioned, as well as which informant mentioned it. In the end, I assembled a paragraph with my findings and their implications for the design.

<table>
<thead>
<tr>
<th>Theme</th>
<th>Keywords</th>
<th>Selected Quotes</th>
<th>#</th>
<th>who</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal</td>
<td>Publicity, visibility</td>
<td>publicity platform, help the scientists be more visible, promote, make knowledge known to the public, popular science, trying to let the world know about all the amazing research, reach the man on the street</td>
<td>9</td>
<td>PM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CR</td>
</tr>
<tr>
<td>Commercial goals</td>
<td>competes with Trondheim and Bergen for the best students, they have gemini.no, more young people to study natural sciences here, and also for the government to see that, I hope that we’ll get more funding for that, then we’ll get connected with politicians and research funds</td>
<td>9</td>
<td>PM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CR</td>
</tr>
<tr>
<td>Quality</td>
<td>recruit the best faculty and students, premium place to study, associated with really good research and good students</td>
<td>3</td>
<td>PM</td>
<td></td>
</tr>
</tbody>
</table>

Table 4 Open-ended, semi-structured interviews: Thematic table for the resulting from Open Coding
<table>
<thead>
<tr>
<th>Nature of system</th>
<th>Solution</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>wanted a standard system, should be one of the open source, freely available CMSs, homemade system would not be good, Wordpress was not good for presenting news</td>
<td>Maintenance</td>
<td>customizations keep on working</td>
</tr>
<tr>
<td>Stakeholder responsibilities</td>
<td>Business Logic</td>
<td>I could do the PHP part, the business logic</td>
</tr>
<tr>
<td>Editors</td>
<td>Communication section, develop and keep it updated, coordinate, read edit and share it, write myself, overview of what’s going on and the strategy, I check everything before, social media</td>
<td></td>
</tr>
<tr>
<td>Tag space separation</td>
<td>The tag spaces are separate from all, create subsites, feed tags</td>
<td></td>
</tr>
<tr>
<td>Tag design</td>
<td>The design was done with the users, heated discussions, they had to be stable, We’ve given input on which tags we would like</td>
<td></td>
</tr>
<tr>
<td>Content priority</td>
<td>Blog quality</td>
<td>The blog posts are not of very high quality, long rambling</td>
</tr>
<tr>
<td>Article quality</td>
<td>Landing on a page with quality content, content full of articles created by people that are editors or writers, substantial article</td>
<td></td>
</tr>
<tr>
<td>Smånytt quality</td>
<td>Confused seeing this teaser and then discovering there are no articles</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>Maturity</td>
<td>No developments so far about extending functionality, what is done is some refactoring</td>
</tr>
<tr>
<td>Upgrading</td>
<td>5 or 6 years in the future, none have been ported</td>
<td></td>
</tr>
<tr>
<td>Addition of roles</td>
<td>No plans to add user roles</td>
<td></td>
</tr>
<tr>
<td>Under tags</td>
<td>Under tags may happen in the next version of the site</td>
<td></td>
</tr>
<tr>
<td>Adding blogs</td>
<td>We have seven, eight blogs and we want more there, we hope we can double it in a year, trying to get new bloggers</td>
<td></td>
</tr>
<tr>
<td>Plans for the future</td>
<td>Facebook campaigns, Work with MailChimp, Contact Alumni, have one new article every day, maintenance and small tasks</td>
<td></td>
</tr>
<tr>
<td>Redesign</td>
<td>Refactoring to reflect what people use, I’m not going to do that very soon</td>
<td></td>
</tr>
<tr>
<td>UC design</td>
<td>Designing with users</td>
<td>Wireframes, Even when designing very closely with users, it turns out that what you design is not what is going to be used, What people express in meetings is not what they actually do</td>
</tr>
<tr>
<td>Experience Level</td>
<td>Editor experience</td>
<td>They don’t know anything about Drupal, Taken a course in analytics, I check that almost every day, I have my own Wordpress blog, MailChimp, Course on Social Media, I don’t know much about these things</td>
</tr>
<tr>
<td>Learning Curve</td>
<td>Cushioned the experience of moving from Wordpress to Drupal, I’ve tried several publishing systems so if you learn one it’s easy to learn a new one, I’ve been working with SharePoint and WordPress and there are some similarities between them</td>
<td>3</td>
</tr>
<tr>
<td>Expectations</td>
<td>Ranking</td>
<td>Show very high up in the search, more intelligent web, search optimization, very curious what it will do, If it makes our site visible I’ll do anything, how to use words and tags to get better results on Google</td>
</tr>
<tr>
<td>Interface Preferences</td>
<td>Editing interface</td>
<td>So much hassle to fix, It has to be user friendly, It’s helpful for people who are not tech savvy because it reminds them of word.</td>
</tr>
<tr>
<td></td>
<td>Help</td>
<td>It works well now, with brukerhåndvisning, it’s easy to find, I do read the help descriptions. It’s really into the point of what we need to know, I don’t like windows popping up, You have to click yourself and read it</td>
</tr>
<tr>
<td></td>
<td>Apprehension</td>
<td>It’s not that dangerous, afraid of deleting a page, motivated if they feel safer</td>
</tr>
<tr>
<td></td>
<td>Restriction</td>
<td>I prefer restricted, erase all channels to do errors, or misunderstanding,</td>
</tr>
</tbody>
</table>

6.1.3 Implications

The themes that emerged from the analysis of the data corpus contributed to delineating the goal of the website as one that intends to attract visibility to the faculty of Mathematics and Natural Sciences. In this way the faculty will be able to promote its research, attract good students and faculty members, as well as increase the possibilities of obtaining funds. The perceived goal justifies the expectations the editorial team has with regards to what the module will achieve in terms of Search Engine Optimization. It also provides a strong extrinsic motivation for adopting the module. An implication of
outlining the goal of the website and the commercial expectations of the users is that after an adequate time period, quantitative data should be generated. This however is part of the future development.

Understanding the nature of the system as one that is open, standardized, and requires little maintenance was important during the choice of the module to be extended. I considered it essential that the module had a high adoption rate, and that it was actively developed and maintained. However, for the reasons explained in section 5.4.1 (forking the module because of the very specific needs of the website), maintaining will be challenging. Yet, recognizing the status of the system as mature and stable may make this challenge less urgent.

Getting insight on the responsibilities of the stakeholders and the different permission levels helped me consolidate the appropriate sample to conduct usability testing with. Realizing that the stakeholders have a busy daily routine led me to create a module that requires little involvement. Moreover their low experience level and their preference towards familiar interfaces guided the design decisions. I endeavored to minimize errors and the feeling of apprehension by providing as many prepopulated fields as I considered necessary (which was proven less than enough during usability testing). Finally, a surprising finding was that the editors were satisfied with the help pages, which was contradictory to the literature guidelines.

Discerning the levels of significance of the different content types helped me decide for which of these I will allow adding meta tags. I started developing and testing around articles, as I consider it the most prominent content type.

6.2 Thinking Aloud

As mentioned in section 3.2.3, concurrent thinking aloud sessions were conducted to test usability. The purpose was to uncover the mental models of the users, as well as misconceptions and difficulties. I also wished to get recommendations for the next increment.

6.2.1 Sample, Context, Task List

In the case of usability testing I sought variance in the sample. I asked the Project Manager and the Project Leader for contact details of active authors and bloggers in Titan, and emailed them with details. The final sample consisted of the two editors, two authors where one was more experienced than the other and one very inexperienced blogger. Having a variation in
user roles and experience level is valuable, as according to the findings of section 6.1.2 a lot of new bloggers will be enrolled into the system. Expecting the bloggers’ thought and action process and designing accordingly increases the possibilities that the module will be used.

The informants were initially asked what kind of work set-up they are used to, and were encouraged to pick a time and a location for the session themselves so that they are as comfortable as possible and the conditions were as close to reality as they could be. They all chose to test outside of the location as most of them work in shared spaces. The sessions took part at IFI colloquium rooms in a span of two weeks. All the tests were conducted on windows machines with large screens and the browser was chosen by the informants.

In the beginning of the session the informants were familiarized with the thinking aloud protocol as a usability testing method, and subsequently presented with a list of representative tasks to complete. The tasks involved creating a content type appropriate to their role, and editing an existing instance of that content type. The instructions were intentionally loosely defined and little explanation over the module’s features was provided, as one of the research questions that guided the sessions was whether they can discover the module and its features themselves, and whether the labels and descriptions were intuitive and understandable. During the sessions I kept notes, audio recordings, and timing. After each session the informants were asked questions about general feedback and certain elements of the module’s interface such as:

1. Prepopulating fields.
2. Expanding and collapsing the blocks.
3. The vertical tab in the left side that shows a preview of the already added meta tags.
4. Help text.
5. Whether they would like more insight into what meta tags are.

### 6.2.2 Thematic Analysis

Correspondingly to the Thematic Analysis phase of the open-ended, semi-structured interviews, I re-wrote my notes and transcribed the recordings after each session. The short individual reflections that were written in order to retain the patterns that surfaced during testing are presented below. The process I followed was analogous to the analysis of the open-ended semi-structured interviews. Thematic analysis does not call for selective coding,
yet, during this round I distinguished Apprehension, Facility, and Redesign Objectives as central themes around which the design should be structured, and the themes Scope, Keywords, and Actorship as the ones that could provide answers to research question 3, regarding the impediments of adopting Semantic Enrichment by non-experts.

Table 5 Thematic table from thinking aloud sessions

<table>
<thead>
<tr>
<th>Theme</th>
<th>Keywords</th>
<th>Selected Quotes</th>
<th>#</th>
<th>who</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apprehension</td>
<td>Guess</td>
<td>So I guess that’s it, I guess I have to think about that, I should start with writing my name I guess, I guess open up these? I guess I shouldn’t write anything here.</td>
<td>4</td>
<td>1, 3</td>
</tr>
<tr>
<td></td>
<td>Don’t know</td>
<td>I don’t know what to do here, I don’t know what that is, I have to read it in order to understand it. If I don’t understand it, I won’t change it. I wondered about Dublin Core, I don’t know what that is. I would have left that blank because I didn’t really understand what that would do for me, What’s Dublin Core</td>
<td>6</td>
<td>1,2,3,4,5</td>
</tr>
<tr>
<td></td>
<td>Uncertainty</td>
<td>I’m not sure I understand this Google news keyword; I’m not sure what this Dublin Core is. Available tokens... I am a little unsure about this, So what do I do now?</td>
<td>5</td>
<td>1,3,4,5</td>
</tr>
<tr>
<td></td>
<td>Making Errors</td>
<td>I don’t want to mess up, Oh no, did I ruin something now?</td>
<td>1</td>
<td>2,4</td>
</tr>
<tr>
<td></td>
<td>Confusion</td>
<td>This is a little bit confusing? The URL for what photo?</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Misconception</td>
<td>Using a comma I can add keywords for Google search?</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Facility</td>
<td>Recognition,</td>
<td>This is very recognizable, is also very familiar, URL looks good. It’s the same, isn’t it?</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Familiarity</td>
<td>I’m not supposed to do a lot here, I see no reason to change it, I don’t see any problems, There’s nothing more I need to do.</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Applying changes</td>
<td>I understand what you mean, I know what to do, that’s easy to understand, nothing difficult here, I get what you mean, Yeah I get that of course it doesn’t why would it? Agh of course I don’t want that? So, this is supposed to make it easier to define for Facebook. It should be y Facebook URL not the name. The same applies for Twitter right?</td>
<td>12</td>
<td>3, 4, 5</td>
</tr>
<tr>
<td></td>
<td>Grasp</td>
<td>I could change the descriptions [...] think creatively about this, It’s actually nice that I get to see all this, it might lower the threshold to publish something, is that understood as a person or as a</td>
<td>6</td>
<td>3, 4, 5</td>
</tr>
<tr>
<td>Company? so I can use for instance my personal page at UiO. I should probably very specific</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contentment</td>
<td>Google+ that’s nice, That’s a nice explanation, this is a good thing because the things you add as a site title or undertitle are not descriptive, this is interesting (Google keywords)</td>
<td>5 4,5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discovery</td>
<td>New</td>
<td>3 3,5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purpose</td>
<td>Use</td>
<td>2 1,4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expectation</td>
<td>High ratings</td>
<td>1 1, 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outreach</td>
<td>Yeah it’s nice to have the opportunity for some extra outreach</td>
<td>1 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Own content</td>
<td>It’s nice to have your post more visible. This is good for me if I get more views</td>
<td>2 4, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Redesign objectives</td>
<td>Redundancy</td>
<td>6 1,2,3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simplicity</td>
<td>Keep it simple because we are trying to teach other writers too., as few as possible, things should be as simple as possible, skip the code if no one understands it</td>
<td>4 1,2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time shortage</td>
<td>everything that’s easy and quick to do is nice, because of time pressure,</td>
<td>2 1,2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepopulating</td>
<td>If I want to go in and put more words it can be filled in automatically, I like the dates prefilled</td>
<td>1 1, 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Permissions</td>
<td>Is this possible both for editors and writers?</td>
<td>1 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Renaming</td>
<td>Change the article author into authors Facebook page, Rename Open Graph into Facebook, what does canonical mean? Grunnlegende meta-tagger, what is a Twitter Card? Dublin Core...is that something secret in Google? Lokalisering can be språkform or something</td>
<td>6 2, 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error handling</td>
<td>Insert a link to the field in the error report</td>
<td>1 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scope</td>
<td>Automatic posts</td>
<td>1 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Twitter accounts</td>
<td>So if someone from the editorial wants to put their account will it show up on their accounts?</td>
<td>1 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Dates

The creation and publication dates are different; will the publishing time be affected?

### External links

Relevant innhold? Is that going to be entered into the article in a way users can see it?

### Actorship

User actorship

The journalist should choose, The ordering is not my choice

### Keywords

Choosing the keywords. Is it the most important thing Google is looking for? What kind of keywords are good for Google news? What is the best way to tag an article? It should be in connection with the feed categories. Nøkkelord in Google, that’s important

### Problems

Bug

It’s quite slow today,

Non-realistic environment

The keyboard is not like the one I work with, I have a Mac so I scroll the other way round. If I take too long that’s why

---

### 6.2.3 Individual Reflections

#### 6.2.3.1 Thinking aloud with Project Leader

The session was impacted by a fault in the Dublin Core Basic module, which I decided to deactivate for the session that was to follow, and errors occurring from another student that was working on the website at the same time. Overall the session lasted about one hour and fifteen minutes.

The main themes of the session were that the informant is occupied with the actual task of writing and editing to a great extent and the act of reading the field labels and descriptions was an overhead to her workflow. That led to the decision to shorten descriptions and have keywords that denote the nature of the input leading the label. Additionally, a change in the naming policy was a consideration as she referred to the URLs as links. Simplifying the module was the informant’s request throughout the session. She was disinclined to add meta tags when the subset was large and when fields were not prepopulated. There were a few occasions where she could not think of an appropriate value to input, namely the Google News related meta tags. I explained that it is not always required to enter a value, and I resolved to make this more obvious in the field descriptions.

Moreover I realized that the scope and boundaries of the module were unclear, as she wondered whether saving an article that uses Open Graph meta tags would entail that it would get published in Facebook, or that
listing an author’s Twitter username would imply that the post would appear in their Twitter page. She also demonstrated interest in deploying meta tags as a means of rating higher in SERPs.

Another interesting outcome was her consideration over the other user roles and the deliberation of limiting some meta tags for bloggers. Overall this session provided important direction for the redesign and the informant was eager to test the redesigned product.

With regards to the employed method, despite gaining valuable insight for redesigning, it often resembled a tutorial as I had to provide instructions to keep the flow. Doing so was against the guidelines for conducting thinking aloud, and more problematic areas could have been uncovered.

6.2.3.2 Thinking aloud with Communication responsible

This session was not impacted by any technical issues since the faulty submodule was deactivated. The session lasted 50 minutes and ended with the informant successfully creating and saving the edited article.

The themes that emerged during the two sessions were largely similar. It was once again recommended to change the naming policies and simplify the available options by eliminating comparable meta tags. Once again I realized it was necessary to clarify the scope of the module as it was unclear whether the publication and modification dates would affect the actual publication and modification dates of the article. Apprehension and fear of ‘messing up’ was a motif that appeared again, as well as the shortage of time. Finally I became aware of the fact that it was unclear whether a Google+ profile was necessary to utilize the Google+ subset.

Overall the informant thoroughly examined the interface, displayed a high degree of autonomy and came up with numerous suggestions for changes. She was also keen in testing the redesigned product.

6.2.3.3 Thinking aloud with very experienced Author

The session lasted 36 minutes, during which the informant successfully created 2 articles, and exposed most of the module’s features, like form validation and Gamification elements. The version of the module that was tested had implemented the changes requested during the first round of the usability testing.
This was the most verbal and autonomous of informants, approaching the ideal thinking-aloud session as described in the literature. He was meticulous with reading the labels and descriptions, as well as inspecting the different options. He truly articulated all his thoughts and had a lot feasible recommendations to offer. He displayed an overall high degree of understanding and feeling familiarity even though he was initially confused by the Grunnleggende meta-data label. He was able to think of examples of the appropriate use of each meta tag. Some of the themes that emerged during the sessions revolved around providing tags that optimally describe the content, adjusting descriptions in order to provide better visibility, changing the labels for Localiserin and Bilde, and making Dublin Core more understandable. Moreover, there was once more confusion on whether the meta tag values are visible to the users. Again, the informant expressed a mindset oriented towards ranking higher. Finally his suggestions about the error message linking to the erroneous field would improve the usability of the module.

6.2.3.4 Thinking aloud with inexperienced Blogger

The informant has never posted before on the website. Having her input was still valuable because Titan plans to onboard a lot of new bloggers, something the Project Leader has expressed during the open-ended, semi-structured interview, and the informant confirmed herself. I found that the session diverged from the thinking aloud protocol as I had to explain all the interface elements, and the scope of my module was not clear to her. The session lasted 32 minutes, under which the informant created a Blog article. I found that disclosing the Gamification elements might be too advanced, so she did not complete the planned activity list. The themes that emerged during the session concerned yet again requesting changes in some labels and naming schemes. Much like the other informants she was interested in higher rankings, but also focused on growing Titan’s outreach. Finally, she recognized the significance of adding proper tags to content.

6.2.3.5 Thinking aloud with experienced Author

The informant is an experienced author in Titan. She was thorough with creating realistic content but did not read all of the labels and descriptions. It took her 25 minutes to create the first article and 8 for the second. She realized the value of providing different descriptions and being specific about tags. Furthermore, she displayed interest on how the entities get categorized in the schema once they are inserted in a meta tag. Just like the others, she was confused with regards to the scope of the module, especially about the
content being automatically shared in social media. She also felt unfamiliarity with the Dublin Core category, and did not notice immediately the status message once the Gamification elements were unlocked. Finally, she connected the value of the module with her getting more readers personally.

6.2.4 Iterative Approach and Implications for Design

An iterative approach to redesign was followed, where I applied changes based on the ‘Redesign Objectives’ theme right after the first round with informants 1 and 2. The results were instantaneous; all the keywords under the theme ‘Facility’ were expressed during round two, by informants 3, 4 and 5 after I took into account informants’ 1 and 2 recommendations. Timing was also improved after each session. It should be once again noted that timing is not an accurate metric due to the overhead of vocalizing one’s thoughts, anecdotal chats during testing, and faults in the code.

The intention was to reduce apprehension, which stems from uncertainty, confusion, making errors and misconceptions. In the same line, an objective was to increase facility; a sense of understanding what the module is about, a feeling of contentment and gradual mastery of the interface. In order to achieve that, I incrementally applied the following changes:

1. Translating the interface to Bokmål.
2. Renaming categories: Open Graph was renamed into Facebook, Twitter Cards into Twitter and Google Plus into Google. Moreover, Basic Metatags was renamed twice with the final name being Grunnleggende Meta-tagger for Søkmotorer (basic meta tags for search engines) and Dublin Core was renamed into Internasjonale Standarder (international standards), as I wanted to clarify what the purpose of each category is.
3. Rewriting labels and descriptions to make them consistent and indicate what the expected input is, thus promoting recognition over recall (Budiu, 2014).
4. Updating the module description to clarify scope.
5. Removing confusing elements like the token browser.
6. Eliminating many of the fields, and eliminating the whole Dublin Core Advanced Category.
7. Revealing some fields only if a relevant selection has been made.
8. Revealing the Google News category once a user has created enough content.
The reactions related to apprehension were not completely diminished, and they can be partly attributed to lack of experience and the unrealistic environment, as well as the tendency of some of the informants not to read the labels. The faults in the code during the first round were fixed for the ones that came, but the performance speed-wise was still not satisfactory.

### 6.2.5 Anticipations and Results

Complete elimination of apprehension was not anticipated because of the reasons mentioned earlier, but also because the subject matter itself can be ambiguous. For instance field experts could not tell when a resource is unique and quality journalism, or when someone is a contributor. I also presumed that unanimity regarding some interface elements would not be reached. However, I expected that implementing the Gamification strategies would provoke more reactions in the ‘Discovery’ theme. As displayed in the table, only 2 of the sessions succeeded in capturing the users’ attention. The intention was to increase the intrinsic motivation of the users, engage them, introduce new features smoothly, and establish flow. So, failing to get noticed was an alarming finding. Of course, Gamification in this case was far from rigorous, and if implemented in the future, it should be thoroughly assessed to ensure that it is applicable in this scenario, that it will not have counterintuitive results, and that it will not entail a case of a ‘hype and disillusionment’.

### 6.3 User Acceptance testing

User Acceptance Testing was described in section 3.4.2. To recapitulate, this is typically the last step before the delivery of the product is accepted or before the product ‘goes live’. By that time the product has undergone unit, functional, integration and system testing. As part of this process, questions regarding correct output, minimum use of resources and load time are asked. UAT however, tests whether the user can actually benefit from the product, and if it is what they asked for. In this case basic/alpha UAT was conducted on the test environment with the Project Leader.

The test cases represented key functionality collated from the information about the users’ workflow, which was elicited in the semi-structured, open-ended interviews. The Project Leader first created an article, and then edited an article and a blog. She was prompted to enter as much information as possible to test the full range of the functionality.

Documentation and timing was kept. It took about 34 minutes to create the new content, and 19 to edit. No faults occurred, but some more ambiguities
were exposed. Additionally, the Project Leader offered some feedback and indicated acceptance of the module. The following points were discussed:

2. Choosing a Facebook page type was unclear.

3. The Date field in the Dublin Core (International standards) meta tag set should be prefilled with the publication and not modification date.

4. Choosing what to enter in the Source (Kilde) and Relevant Content (Relevant Innhold) fields from the Dublin Core (International Standard) subset was unclear.

5. Regrouping the fields in each subset not alphabetically as they were, but in a way that places the most prominent fields on the top.

6.4 Conclusion

This chapter detailed how the qualitative data that underpinned the development of this module was generated and analyzed. The analysis guided the design and provided the answers to the research questions, which will be examined in Chapter 7.

Conducting open-ended, semi-structured interviews provided insights into the nature of the website, the workflow, mental models and expectations of the users. Having this knowledge aided with selecting an appropriate module to base development on.

The concurrent thinking aloud protocol tested the usability of the high fidelity prototype. Ambiguous areas and misconceptions were uncovered, and a lot of feedback was offered by the users, all of which were actionable design guidelines for the next iteration. The design objective was reducing apprehension and increasing facility. The user sample displayed variance in the level of experience, which is crucial since Titan plans to onboard a lot of new bloggers. A challenge of employing the concurrent thinking aloud protocol was that not all of the users continuously verbalized their thoughts. I had to intervene and provide clarifications. The situation was also unnatural, both because of the location, and the equipment. Seeing immediate results with regards to facility convinced me that the method is robust and flexible even for inexperienced researchers.

It is however not enough on its own because more issues emerged during acceptance testing. Finding representative tasks was in all cases straightforward after applying thematic analysis. Qualitative data was the focal point of this study, but quantitative data that could be gathered from logs and A/B tests would benefit the usability of the module.
7. Discussion

This chapter discusses how the research questions were answered, which challenges were faced, which assumptions were made, what are the theoretical and perceived benefits of Semantic Enrichment, and which steps can be followed in the future.

7.1 Answering the Research Questions

The aim of the thesis was to explore the semantic enrichment of the content of a CMS driven news outlet. In this context the following research questions were asked:

7.1.1 What are the requirements for developing an understandable module that will enable Semantic Enrichment for users with no expertise in the field of the Semantic Web?

The word ‘requirements’ is usually associated with gathering and documenting the needs of the target user group early in the design process. Contradictorily, this work started with assessing existing modules that could serve as a high level prototype. What followed were open-ended, semi-structured interviews that would determine the experience level, workflow and mental models of the users. Thus, the requirements were not directly elicited, but collated after applying thematic analysis to the qualitative data corpus. These requirements directed the first iteration of modifications and additions of the high level prototype. Additional requirements emerged during usability testing, and even during user acceptance testing.

The users requested fine grained permissions, reducing the learning curve by keeping the interface consistent with what they are accustomed to and erasing all channels for errors. They also asked for removing technical features like the token browser, fields that they considered redundant and explanatory elements like tooltips – a surprising finding since it opposes the literature’s guidelines. Field prepopulation, another common request, seemed to draw the attention of some informants but distract others. The first group namely expressed the importance of having variation in the descriptions, while the latter did not pay attention to the fields that were prepopulated. Comprehensive renaming was another necessity, and what was proven to be most effective was placing the keyword that indicates the expected input first. In the same line, users expected that the subset name reflects ‘what it does for them, what it is used for’. Finally, I had to clarify the scope and
boundaries of the module to most of the informants. I updated the module description to inform the users that by saving a piece of content it will not for example get automatically shared on their Facebook profiles. Besides the requirements from the users, it was important that the output conformed to the standards.

7.1.2 What are the technical challenges faced?

Researching and implementing this module posed a number of challenges that span multiple territories:

- Regarding semantic enrichment and meta tags as a field: As mentioned in the introduction of this thesis, many (Konstantinou et al., 2010; Shadbolt et al., 2006) argue that the Semantic Web has failed to deliver because even though the fundamental technologies are ready to be deployed outside academia, non-experts have not taken advantage of them. Yet, one can observe advancements and thus instability in the lowercase version of the field. For example, Google stopped supporting the Authorship meta tag set which aimed to bring prominence and credibility to authors. At the same time they keep launching new tags. The most recent addition was that of the ‘Local Source’ tag (Morehead, 2016) which aims to surface content from local news publishers in Google News. This means that the meta tags that are currently supported need to be reviewed and maintained periodically. Working with a field in a state of flux is a challenge in itself.

- Regarding the design process and the design: Being based on an existing module directed and shaped the design and did not leave much room for exploring alternatives. The module itself had a number of shortcomings and performance problems that were infeasible to fix within the set timeframe. Another dubious issue is the removal of perceived redundancies. This was one of the most common requests among the users, but even after some rounds of removals, fields like the title and the description appear throughout all of the subsets. This was an intentional compromise in order to keep some level of consistency so that each subset has approximately the same type of fields. A different approach could have been used where fields are grouped by expected input, but in that case, the request of the users about knowing what a subset has to offer them is no longer satisfied. Finally, assumptions were made concerning which output is correct. I namely followed meta tag standards such as the OWMS profile that is used in governmental contexts.
• Regarding Drupal’s learning curve, timeline estimation and shifting gears: Early on in the development process I became aware of the steep learning curve of Drupal. Even though I have taken courses on Information Architecture that used Drupal as a tool and had some knowledge on PHP, I found that the functionality I wanted to develop could be implemented in more than one ways, some of which may not have been proper. Running on technical problems that took me a long time to resolve or compromise with, led to deviations from my original time plan and research focus. I namely aimed to follow a more quantitative approach like the one described in the future steps. Undoubtedly this would have led to a less strong focus on the users. The scope was once more drawn to another direction by implementing a lightweight approach to Gamification. Narrowing down the research aims earlier would have allowed me to proceed further.

• Regarding the final module: The module adds an overhead to the performance of the site, but keeping it disabled results in the meta tags not being generated. Another difficulty will be keeping updated with the original Metatag – a disadvantage of forking and refactoring.

7.1.3 What impedes the adoption of Semantic Enrichment by non-experts? Why is the Semantic Web underused?

Chapter 2 confirmed that the tools for bootstrapping the Semantic Web have reached maturity. It is however still the case that the vision is not realized. This thesis attempted to rectify this situation, and provide an answer as to why the phenomenon occurs. At the outset, one has to accept that a lower-case approach is obligatory for the target users. Even after assessing and choosing the most complete and appropriate module, I found that it cannot be used out of the box by the target audience. The steps taken to amend the situation yielded positive results but not full actualization. According to the empirical findings, this is why:

Lack of time and expertise are innate and expected attributes of the users that the design aimed to cater for. Implementing their suggestions did indeed lead to lower apprehension and higher facility, but even during the latter phases of usability testing, their understanding of the module’s scope was vague. Visualizing the sphere of different services and how these can connect to each other is not rudimentary.
The users also held misconceptions regarding the appropriate input in each field, not only because some skimmed over the descriptions, but also because classifying external resources may be ambiguous even for field experts. For example most of them struggled with defining ‘unique and quality journalism’, or with deciding what kind of keywords are suitable.

The users were also divided on their take on ‘actorship’. Some of them wished that specific fields are not prepopulated because this is when the fields are ‘brought out’, and this is when they get to choose for themselves, while others claimed that once the field is prepopulated it draws their attention and makes it more likely that they input a more appropriate value. This was the case with the Category fields. While the standard Drupal fields allow more than one values, the module expects only one so the users should choose consciously. There was no univocal conclusion on whether the field should be prepopulated with the first out of N choices, or whether it should be left empty.

Gamification strategies, especially scaffolding, aimed at increasing the intrinsic motivation of the users, and progressively disclosing more features to achieve better flow. I expected that the ‘Discovery’ theme would be more prominent, but some of the users seemed to notice only after my hints. One could argue that using Drupal’s status bar to notify the user when ‘a level is unlocked’ is too subtle, and that a pop up would be more effective. Yet, this approach would collide with the findings from the open ended, semi structured interviews, where users expressed dislike towards pop ups and similar interface elements. Developing a fully fledged gamified module with social and competitive aspects could possibly engage the users if it would not come with a significant learning overhead. Implementing and assessing this statement this was out of the scope of the thesis.

To conclude, even when following a user-centered approach, issues like visualizing scope, the extent of actorship, and accomplishing engagement still persist. One academic year is not enough to develop a solution that lets users achieve ‘flow’. In the same line, refactoring may be necessary in the future, as indicated by the following utterance from the open-ended semi-structured interviews: “Even when designing very closely with users, it turns out that what you design is not what is going to be used”.

87
7.2 The perceived and theoretical benefits of Semantic enrichment in this use case

Since the module has not been launched on the production environment for an adequate amount of time, no quantitative claims regarding the benefits can be made. The users however perceive a lot of benefits which are also backed by the literature:

1. Increasing visibility and content attribution both when it comes to the SERP (Search Engine Results Page) and social media. Especially the Rich Snippets and Knowledge Graph Cards can enhance the appearance and coverage of a website as they enable visual elements and provide answers regarding the entity, not just links (Sullivan, 2012).
2. Tailoring how the content appears when shared on social media.
3. Defining canonical URLs disambiguates and uniquely identifies content, thus eliminating duplicates.
4. Categorizing entities can give rise to various applications like personalized recommendations and semantic search.
5. Since several meta tag sets are connected with commercial actors, this may give content providers a stronger incentive to perform semantic enrichment. A direct consequence would be the ability to connect entities across different websites and set the ground for linking data.

7.3 Future steps

As it becomes apparent, there is a lot of space for improvement in the following areas:

- Implementation-wise: Stricter validation functions can ensure that the input is an active Facebook, Twitter or Google+ profiles with API calls. Since this may impact performance it can be an additional choice for users. For the time being regular expressions are used to make sure that a URL is well formed and contains the words facebook or google. AJAX form validation may also be useful assuming that users do not read thoroughly the descriptions. Performance issues should be investigated and resolved, and finally, instead of having a forked module, the features that are not specific to Titan can be generalized into a patch. Of course, issues in Metatag’s queue can also be tackled. In addition the module can be extended with a submodule that supports an ontology/vocabulary of choice. Finally, a fully
fledged Gamification approach with social and competitive elements like badges, levels and leaderboards can be pursued.

- This thesis has taken a qualitative perspective. The module should be studied quantitatively to confirm or disprove the perceived and theoretical benefits. Logs of usage can evaluate task frequencies and sequences under active use and capture errors and productivity (Shneiderman et al., 2013, pp. 145, 155). This is in line with the findings of the open-ended, semi-structured interviews: “even when designing very closely with users, it turns out that what you design is not what is going to be used [...] you accumulate data for a year, you analyze that data and then you basically just write some software to do the refactoring of your tags to reflect what people actually use”. Moreover, different layouts, multistep forms, and features for which unanimity was not achieved can be tested with A/B tests (Nielsen, 2005) in order to find out which version perform best, and measure actual user behavior. Studying the output of Google Analytics, Facebook Insights and Twitter Card analytics\(^{25}\) can help with tracking how meta tags drive clicks and shares, and if visits increase. Enriching some instances while leaving others untagged and comparing the number of visits would serve the same purpose.

- Even though the methods used to gather and analyze the data corpus were revealing and flexible, they were not enough. More qualitative usability testing can be beneficial since users were uncertain about some interface element even during acceptance testing. Field tests and observations that put the interface to work in the natural environment for a fixed period can uncover more of these ambiguities. Shneiderman et al. (2013, p. 154) propose that the period of field testing that follows successful acceptance testing not only refines the user interface but can also improve training methods and help materials.

\(^{25}\) [https://dev.twitter.com/cards/analytics](https://dev.twitter.com/cards/analytics)
8. Conclusion

The contribution of this work is twofold. First, it explored Semantic Enrichment, its perceived and theoretical benefits, as well as the technical challenges and impediments. Secondly, it developed a solution aiming to rectify the challenges and impediments in the context of Titan, the news outlet of the faculty of Mathematics and Natural Sciences in the University of Oslo.

Initially the target user group and their needs were identified. The objective was catering for non-experts who require consistency, simplicity, and effortless data entry. Additionally, providing explanations and error recovery, easing the learning curve, not causing cognitive overload, and allowing for fine grained access and sustained engagement were fundamental design goals.

The background and state of the art on the field of Semantic Web were outlined. Focus was given to the features of the vocabularies used in the module as well as the adoption rates of different annotation formats. A review of the core and contributed modules that deploy semantic technologies in Drupal was also given. Meta-tagging was presented as a lower case semantic technology that allows content providers to add extra information to a web page for the purposes of determining content relativity, site niche, visibility and attribution.

The procedure for conducting research, gathering and analyzing the qualitative data corpus was presented. I followed an approach where user involvement and feedback is central. However, the design process did not fully comply with the typical life cycle model for user centered development due to time restrictions and choosing Metatag, a Drupal contributed module, as a the base of the high level prototype. In order to gather data and test usability, open-ended semi-structured interviews, concurrent thinking aloud protocols and user acceptance testing were employed. The data was examined using thematic analysis with open coding.

Prototyping followed several guidelines from the literature. These guidelines directed the design of forms – the main visual element of the module, and proposed Gamification strategies that increase intrinsic motivation and sustain engagement. Key points of the guidelines were consistency, grouping and sequencing menu items, providing meaningful and familiar to the users titles, labels and instructions, as well as widgets to increase selection speed, marking required fields, and generating error messages that indicate what
caused the error and what the permissible alternatives are. Moreover I designed for a single point of entry, hid as many technical concepts as possible, reduced complexity by displaying only the relevant to a selection fields, enabled advanced permission control, and automation. Finally, I implemented scaffolding by unlocking functionality once the user gains enough experience.

During implementation I identified which augmentations and modifications were required to make Metatag fit for the needs of Titan. The workflow, languages and toolset when working with Drupal were described and a list of good development practices was presented. I outlined the iterative phases of development which consisted of singling out the relevant and supported subsets and meta tags, making the output conform to the standards, renaming and adding default values, widgets and translations, regrouping, prepopulating and adding Gamification elements.

Development, data gathering, and analysis were intertwined processes. Applying thematic analysis on the material from the open-ended semi-structured interviews helped me discern the goal of the website, the workflow, mental models, expectations, apprehensions, and motivations of the target users, as well as uncovering ambiguities and misconceptions. These themes guided the selection of the starting module, consolidated the appropriate sample for usability testing and determined the users’ representative tasks. I also came across findings regarding the help material that contradicted the literature. During usability testing I sought variance in the sample since new bloggers are about to be onboarded. The most important themes I identified were apprehension which stemmed from uncertainty, confusion and misconception, and facility which consisted of recognition and familiarity, understandability, mastery — expressed with a willingness to apply changes, and contentment. Following an iterative approach where changes were applied right after each session round, gave instantaneous results; however reactions related to apprehension were not completely diminished, and some expectations regarding Gamification were not materialized. Finally, I conducted User Acceptance Testing to verify that the module fulfills the users’ requirements. UAT exposed some more ambiguities which became the objectives of the final redesign. The findings from Usability testing and User Acceptance Testing provided answers to all three of the research questions.

One may question the value of the module besides the theoretical and perceived benefits. To start with, having semantic web experts providing
enrichment or enrichment schemes, is not a choice because the site offers a unique feed of content on a daily basis. One might argue that since most of the fields are prepopulated, it could be possible to generate the meta tags without human agency. An answer to this argument is that it might be desirable to serve different titles, descriptions, images etc. to different channels. Additionally, meta tag fields get prepopulated based on the standard fields of the content type. In some cases a direct counterpart does not exist, so the users have to think for themselves. For example blogs don’t have a subsite field, and such a categorization is required by OGP.

Some reflections from the process are that further qualitative studies are required to examine situated use and eliminate the remaining difficulties the users face. Moreover, it would be interesting to test and assess Gamification in a fuller extent. Overall, more time is required in order to undertake and fully succeed in such a project. Quantitative studies may also help with choosing between competing designs, but even if user-friendliness is achieved, it is still critical to resolve the remaining performance issues. Being based on an existing module meant that I could leverage on functionality that I could not have implemented myself. However, the design got shaped too soon and several compromises had to be made. Finally, another lesson I have learned concerned the importance of narrowing down the research scope earlier.

Even though the Upper Case Semantic Web technologies have undergone standardization and maturity, the lower case variant — that is more likely to be used by non-experts — is still evolving. Semantically enriching content is a field with both commercial and academic value, and a ‘niche’ where progress is still to be made. I look forward to seeing the next steps implemented, and the quantitative studies proving the perceived and theoretical benefits of semantic enrichment. I hope to be able to keep contributing to Titan.
Bibliography


Cotton, P. (2013). WG Decision to remove Microdata from HTML 5.0, remove JS API, continue HTML Microdata as a separate spec. Retrieved from [https://www.w3.org/Bugs/Public/show_bug.cgi?id=22783#c0](https://www.w3.org/Bugs/Public/show_bug.cgi?id=22783#c0)


Linked Data in HTML. Retrieved from http://rdfa.info/


Shneiderman, S. B., Plaisant, C., Cohen, M., & Jacobs, S. (2013). Designing the user interface (Fifth ed.): Pearson Addison Wesley, USA.


Appendix A: Open-Ended, Semi-Structured Interviews

Interview with Product Manager (Gisle Hannemyr)

The interview took place at the informant’s office at 10.02.2016.

What are the goals of the website?

The goal of the website is to be a publishing platform for the Mathematics and Natural Sciences faculty. The target audience is alumni, businesses, prospective students, and generally to show that this is a faculty producing interesting research, to promote it as a premium place to study, and to collaborate with, and just to create good will for the faculty. There are commercial goals to the extent that the university environment in 2016 is commercial. The UiO Science department competes with the University of Trondheim, the University of Bergen etc. for the best students, so it’s clearly commercial. I mean, for a university to be successful, it has to be able to recruit both the best faculty and the best students, and Titan is intended to help in that direction. But there are no other commercial goals. It is not going to be used to make money directly. It is not going to be used as a platform for advertising. All the economic goals are indirect by promoting the faculty.

Could you describe the project timeline? What are the plans for the future?

I was approached by Gunhild Haugnes who is the Project Leader. Morten Dæhlen, who is the dean of the faculty, and I think it was actually Morten who had the idea but I’m not sure about it.

Gunhild is employed full time by the faculty as a publicity person. She handles the publicity for the faculty along with her colleague Ida, but I am not completely certain about the organizational structure up there. The idea was basically because Trondheim had their own science website and the University College had khrono, we wanted something like that as well, and that was the inspiration.

I was basically invited into the project because they wanted somebody who was already employed here and who knew something about web design. There are not many people here working on web design. USIT is of course
doing it, but USIT is using this platform called Vortex which they created from scratch themselves, and my impression was that these people who initiated this didn’t want Vortex. They wanted something else. They wanted a standard system. The other person who was in the pre-project that took place in the spring of 2015 was Kristin Skar, who is managing the department’s computers.

Morten has his own blog, and Morten didn’t want to use Vortex so the blog was based on Wordpress, and Kristin had actually made it for him. So, it was Morten, Gunhild, Kristin Skar and I. And it was already decided that it should not be an inhouse platform like Vortex, it should be one of the open source, freely available CMSs. And there were really two candidates; Wordpress and Drupal. And Kristin was very unhappy with Wordpress. I think Morten has hired some outside web Development Company to customize it for him. And these people didn’t know what they were doing. So they hadn’t really customized it. Wordpress comes with a new security release every month. Every time a new release came out Kristin had to reverse engineer all the traces this outside company has done to Wordpress and merge them into it again. I’m not a Wordpress expert and I don’t think that Wordpress is that bad really. I think that if you know Wordpress and you know what you’re doing you wouldn’t have to do that. When she asked me if Drupal is like that, I said no, Drupal comes with a security release every month or so, but if you do it right there is no overhead. You just install it and the customizations keep on working provided you know what you’re doing. So Kristin wasn’t happy with Wordpress and I was happy with Drupal so that’s why Drupal was chosen.

And then they asked me if I could do it and I said I could do the PHP part, the business logic. I am not a front end designer, so I suggested that they use some outside agency to create the front end design for it. And then I could take that front end design and create the business logic in PHP and SQL and make it into the real site.

So, first they tried to hire the same front end designer that khrono has used. Gunhild liked the khrono site very much, which is actually written in Drupal. This designer was familiar with Drupal and has proven capable of making a good front end design that was suitable for Drupal. So, we tried to hire him and then the university interfered saying ‘well you can’t just hire this Swede from out of nowhere’. We have this blanket agreement with Netlife Research, and all outside web development has to be done by them. And I asked do they know Drupal, and they answered ‘yes we are very familiar with Drupal,
that’s no problem’. Then they were hired and first thing I discovered was they didn’t know a thing about Drupal, really didn’t know how to do it, so we had to teach them, and still they actually did it the wrong way so the front end design was basically just static HTML, so I had to do the integration with Drupal. That’s why there are some rough spots in the front end design. If you look at it on a tablet and you turn it sideways some of the CSS breaks. And we’re out of money to hire an external consultant to fix that. It’s not Netlife Research, because I don’t think they were right for the job, but someone who actually knows Drupal and front end design, but I don’t have the skills; I would spend a long time doing this so it’s much better to hire somebody else.

I created the backend design and the business logic and custom content types. All the major content types, Article, Smånytt and Blogg, are just trivial extensions of the content types that are in Drupal core, but I created some other custom content types, for instance Quote of the Day and Researcher of the Week, which are created outside the Drupal node system and the Drupal entity system. 95% is just standard Drupal stuff.

So I didn’t teach in autumn; I teach in the spring. I spent autumn putting the site up and I think we launched on the 12th of December 2015, and at that point all the planned functionality was ready, so that what’s the site’s supposed to do and there haven’t been any developments so far about extending functionality. What’s been done since then is some refactoring. That’s why I didn’t have the opportunity to synch it last week. Because they request that the publishing workflow for Smånytt should be changed. The articles initially are unpublished and then the editor has to oversee and click the ‘published’ button and Smånytt is just a trivial small news item so there is no editorial oversight in the workflow, and the users just said, ‘well, it’s confusing that these things operate differently’. So we did 2 changes. One, we made both of them unpublished by default, and two we empowered the writers to actually publish. So there is no editorial oversight on Smånytt but they are published by the authors, they are not automatically published.

So, speaking of few tweaks like that, I still have a list of 25 items which are small bugs which need to be looked into, which I do in between other things. I am teaching this semester so it’s very slow progress. But none of the bugs are critical. I don’t think that if you don’t know where to look you will have problems. I don’t think there are any plans about any significant extensions. What you see now is certainly what will be there during the duration of your thesis, and probably also for a long time after.
What is the workflow for the three main content types?

The editors, Gunhild and Ida are allowed to do anything, and the writers are allowed to write and continue to edit, so everything now, except blogs are unpublished by default.

So the writers are working on an article and they might stop working on it, return to it days later to add more items or photographs and so on, and when the writer is happy, I’m not sure if they ask Gunhild for permission or if they just do it, but they push the ‘publish’ button and it’s published. And for Smånytt that’s exactly the same workflow and permissions. Except for Smånytt they don’t have to ask Gunhild for permission because they were automatically published before. Gunhild doesn’t interfere with blogs at all.

Each blogger has the permission to create, and to continue editing and to delete his own blog posts and nobody else really interferes with that. In some of the articles there are more collaborative things, like more than one author being involved. And the blogs are automatically published because there is no overlap between the blogger and the writer role. So that workflow is different. But the bloggers can unpublish if they want. There is a tab on each item which you can click to publish or unpublish. And bloggers have access to that.

Also tagging. I think the main person responsible is the writer. The tag spaces are actually separate for all. This is one of the design changes we did since we deployed. We started out with just one tag space. In the beginning when you clicked in the tag for mathematics you got back all the articles for mathematics, all the blog posts for mathematics and Smånytt for mathematics. Because the blog posts in general are not of very high quality - that happened almost immediately after launching in December - we separated the blog posts in a different tag category. You will find the same tags in the blog posts as in articles, but because they are different vocabularies in Drupal terms they end up in different views. That means that no blog post will be shown if you click on a tag in the front page. And then just a couple of weeks ago we did the same change for Smånytt. Smånytt also has its own category.

You know the Drupal front page; you have this red area in the top right corner where there are all the main tags in the front page; astronomy, mathematics and so on. And you really wanted that people who click those in the front page are going to be landing on a page with quality content.
That means content full of articles created by the editorial, people that are editors or writers. It shouldn’t be Smånytt because people were just confused seeing this teaser and then discovering that there are no articles, just 5 lines long. User expectance is, that here is a substantial article, not short notes. And they certainly don’t want a blog post with long rambling running on sentences. Some of our bloggers, I won’t mention names, but they write like that.

In addition to those tags which basically are used to control topic oriented subsites, then the major tag, astronomy mathematics and so on, actually is used to create subsites.

There are also feed tags. They are not publically visible. They are set by the editors, not the writers. But the editors might train the writers to use them, because Titan is also a newsletter based on MailChimp. The MailChimp API is based on RSS feeds. So basically you tell MailChimp I want this RS feed to be a weekly newsletter. There is one feed for each institute under the faculty, plus there is one feed for the centre for excellence in research. And those tags don’t appear, they are hidden from normal users of the site, they just generate newsletters. So you basically can go to Titan and say I want to subscribe to the newsletter from the Informatics Department and you will then get all the news once a week tagged with the informatics feed tag. Some of these tags are actually the same as the subsite tags. Astronomy for instance is for the department and the feed. But for instance there is no informatics subsite. The informatics material goes to the technology subsite.

After some time it might be a good idea to have one year of data to actually find out what tags are used, what tags are not used. What sort of synonym relations, synonym rings, all these you do in information architecture to make sure your tags are sensible and that they are useful for finding stuff and navigating. But if there doesn’t already exist some vocabulary like Dublin Core for instance, for tagging, I haven’t had good experience designing it. Even when designing very closely with users. It turns out that what you design is not what is going to be used. It is a bit like shortcuts. Landscape architects can map where they want people to walk, and then they design it and then you look out of the window and see that grass is absent from spots where people walk. That’s the way it is with tags as well, but with software it’s much simpler to redesign it. So you accumulate data for a year, you analyze that data and then you basically just write some software to do the refactoring of your tags to reflect what people actually use.
Was it only you that designed the taxonomy on the current site?

No it was done with users. I didn’t actually design it, I was present at the meeting where it was discussed and took notes and said ‘well this is what I could gain from the discussion, does this look all right to you?’ The users did some wireframes. There were some very intense discussions about tagging. For instance the innovation tag which is one of the main subsite tags was not there in the beginning, and then some said innovation is very important to the University, we need to have an innovation subsite. So the design was done with users, but now I have two months of data and I can see that the users already don’t use the tags consistently. Some tags are overused and some tags are overloaded, so I already now understand that at some point I need to do some refactoring, because what people express in meetings and what they actually do is not necessarily the same thing, but I’m not going to do that very soon, probably after you hand in your thesis, and perhaps even the thesis may provide valuable info about tagging, because there is a close relationship between the native tagging which we do for our own purposes and the semantic enrichment tagging which you do for external purposes.

How many meetings did you have?

A lot. I don’t have a count, but except for holidays there has to be at least one every week, so probably thirty or so. It started out in May, there was a break during the summer and we started again in autumn. The progress is so slow now, but the users want to have weekly meetings, so I’m happy to go along with that.

Are you thinking about upgrading to Drupal 8?

No. At one point of course, but it’s probably 5 or 6 years in the future. The reason is that Drupal 8 has been released almost none of the modules that Titan relies on have been ported. I’m not going to sit down and do the porting for all those modules myself, so I’ll stay to Drupal 7 until all the key modules exist in Drupal 8, or some other workable solution exists. I mean, a couple of the modules, I’m actually a maintainer of, Anonymous Publishing for example, and I haven’t started porting it. I just said that if somebody wants to do it I’ll make them a co-maintainer but I don’t have the time to do that.

Are there any SEO strategies in place already?
Not as far as I know.

Did you follow any design guidelines? Did the front end developer follow any design guidelines?

I don’t think they did. They created the wireframe, got that approved by Gunhild, not from me, and then they implemented that in static HTML and static CSS, no PHP no business logic at all, and JavaScript of course was also included. They probably followed whatever guidelines exist in Netlife Research.

Who is responsible for the social media accounts?

Mainly Ida I think, not Gunhild, but you can ask the editors. It was created by the editors not me and I think Ida is the one maintaining it, mainly by clicking on the Facebook button on the Titan website. The reason I know about this was because there was a bug in it (Discussion and display about a bug follows – it turns out it was due to Titan caching. Content was only available to logged in users. Facebook’s scraper is an anonymous user and it was denied access until the cache was rebuilt.)

Are you using any analytics tools?

Google Analytics. I am not using it. I set it up and Gunhild is using it. Actually she’s even gone through a course.

What’s the approach on mobility? Is it only the front end design that is responsive? Mobile deep linking?

As far as I know only the front end is responsive. It is based on Twitter Bootstrap. What are you referring to with the phrase deep linking?

(I explain about the specifics of URIs launching directly mobile apps, and how the metatag module allows mobile deep linking tags).

If this deep linking thing could be used [...]. I believe that 50% of the users are actually coming to Titan through mobile platforms according to Google analytics.

There is a production environment and a development environment?

Yeah, there are completely separated but kept in sync with rsync. Basically what I do to sync them is I copy the entire file tree for configuration from
one to another with rsync and then I swap in the settings files afterwards because the settings also travel along with the regressive rsync, but you have to replace it with a custom settings file, otherwise things like the Drupal root may be wrong, and the name of the website. There are actually 4 machines involved: Production webserver and database server, staging webserver and staging database server. To sync the databases I just do a MySQL dump, copy, and do an insert. I have to run two scripts. So what I’m doing is debugging and looking for new features and so on, on the staging server, and when I’m happy with what I created, it’s ready to be deployed. I always sync from the production to the test server. I usually work on one feature at a time. When one feature is ready instead of doing a sync, I put it in the git repo on Drupal.org and then I install it like installing any other module on Drupal from the git repo. And that means that the chances of breaking production are actually smaller.

Any plans to add user roles?

No, not currently.

Did any of the users have training in Drupal? Do they just know the basics?

They don’t know anything about Drupal. They know how to create content. In Titan there is a WYSIWYG editor. I hate WYSIWYG. I think they are slow, messy, they break your markup. But there was no way of getting that user group to use anything other than that, because most of these people had prior exposure in creating content in Wordpress. That editor is just a JavaScript library. That sort of cushioned the experience of moving from Wordpress to Drupal.

Once this module is developed, how can it be maintained?

If your module is successful, because you are the author you actually have a copyright and can decide what you want to do with it. What I would suggest is that if it works, you put it up on Drupal.org, if you’re interested in going through that horrible process of getting a vetted user to demonstrate your coding skills. If you’re not interested I can put it up there, make it a full project immediately and make you a co-maintainer, essentially the sole maintainer. And then suggest that the ownership is handed to you. One way of becoming a project owner is applying in the queue, but it's a horrible user experience. Of course with some coaching is less horrible. The second route is that you are given custody of one module. Then you don’t have the right to
promote other projects, but the full ownership of the module you created. Or if you’re not interested in maintaining it, I put it up on Drupal.org, making sure that you are credited, and say this one is up for adoption. Or, you can share it with nobody. Titan can have it but you can sell it.

Interview with Project Leader (Gunhild Merete Haugnes)

Tell me about your duties in the university, what is your typical working day? Is Titan a full time employment for you, and if not, how does it fit in your working day?

It is almost full time as I work in the communication section of the faculty and sometimes I have to do other things also, but last year Titan has been the main work to develop and now it’s also to keep it updated every day with new articles and so on. I’ve been part since the start, the development with Gisle and the service platform, the technologic in the background. I also use my time to find people to write, and get to know the scientists of the faculty, and now we are two journalists working with writing articles and also some scientists and other communications people from the faculty do that too. So I have to coordinate all these things. So we have at least one new article published every day and I have to read it and edit it and share it in social media, and we’re still working with development like putting new technology in place and that kind of thing so that we can spread it and also have some campaigns in Facebook and other social media. I write myself also, articles, especially about IT actually and innovation and technology.

Is that your background?

Yeah, I’m an electronic engineer from a long time ago and I also have economics and I’m also a journalist for over 20 years in technology and finance.

Who is it that takes the initiatives in this project?

You can say I had the idea. When I started at the university at the communications section about 3 years ago, my job was to help the scientists be more visible, but very soon I found out that the university did not have a good channel to get all the news out because there were too few articles. Information about exciting projects was not easy to find, and in many different places. And I thought we needed a new site to write about science, news and also activities at the faculty. That’s also because of the competitor NTNU in Trondheim they have a similar one, gemini.no, and we have to
compete with them and be more visible. Now we have a base, someone to write and someone to edit, but we want to spread this out, so we have a newsletter and also some campaigns on social media and SEO and semantic web to make it more visible. It’s not just me taking initiatives, because of the dean, Morten Dæhlen, he has thought about this before and also owns a blog site, and when we discussed it in the communication section, we thought why don’t we start this. That’s why we also have blogs there. So at the moment we have seven-eight blogs and we want more there. And of course Gisle was very enthusiastic about Titan he wanted to develop it. So that is the background for starting it.

So now that you’ve mentioned Trondheim, what is the main goal of the website?

In one way Trondheim is on the same side as us, because they work with making visible natural sciences and technology, and of course making knowledge known to the public. The university’s main goal is to develop knowledge, and not only for the students and the scientists but we also to share the knowledge with the people that are financing all this, and the researchers who are working here get smarter themselves so it’s building up the reputation to show all the exciting things happening here. And of course we want more young people to study technology and natural sciences here and also for the government to see that. And also for the scientists. Before we started this project, I visited all of the institutes in the faculty and I tried to learn more about what they were doing, and they wanted to talk about their science but they didn’t know how. So it is also for them because there is a threshold to have science published. Of course there are scientific magazines also, but this is for popular science. I also think there are some internal purposes because the scientists are working in very different fields and they don’t know what the scientist in the room beside them is working on. I find it’s nice to get an overview of what the faculty is actually doing.

What have you learned from working in the project?

I’ve learned a lot about science I didn’t know before; I met a lot of interesting people. As I said I’ve been a journalist for more than 20 years, but the people I like to interview the most are innovators and scientists. I like that they want to create something, and curiosity, and some kind of passion about what they’re doing. So that’s the reason why I wanted to start at the university, because here there are a lot of scientists and a lot of innovators actually. I like to think I am a bit of an innovator myself, of
course this is a small site, but I like building up things. I also like to work with storytelling.

What is your exact role and how does it relate to the other roles in the site. Do you perceive any hierarchy between the roles?

I am an editor so I have the daily responsibility for the content on the website, but the leadership has the formal responsibility for the site. The bloggers, they themselves are editors for their own blogs. I also write stories and I am the project leader, coordinating everything. Gisle is the IT specialist. I have to have an overview of what’s going on and the strategy.

Is it the case that everyone’s material is streamlined and has to go through you before being published?

I read all the stories in the front page and publish them, because I try to publish one case every day, but from time to time I say, if I don’t have time I tell to the writer, ok just publish it. But usually I do it. But for the Smánytt I tell them, if you find anything interesting just write it and publish it. I usually look at it afterwards. But usually I check everything before.

What do you know about SEO?

I don’t know much about these things, but of course it’s like being visible for the search engines like google and bing. So if people are searching on specific fields then it would be nice that they would be shown very high up in the search of course.

What do you know about Google Analytics?

I have installed that one and I check it almost every day and I see which articles are most popular and where the visitors come from and how much escape rate. So I check out that often to see what kind of content people read, what kind of stories. So I want to have the statistics of that after some time again.

Are you thinking about modifying the content based on the analytics?

If something doesn’t go well in the analytics we might write about it anyway. We are not looking for clicks because we are not an ordinary newspaper, but of course we have to learn from it and say oh people like that kind of stories, and try to learn.
What do you know about Drupal?

Of course I know it’s an open source system for publishing and we had a discussion on what kind of system we wanted on this site and Gisle was up for Drupal. He was very convinced that open source was the way to go because at UiO we have a homemade system that would not be good for what we wanted to make. We were talking about WordPress and it was mainly good about the blogs but not good for presenting news.

When you are writing an article, do you find that you first type it in an editor and then copy and paste it into Drupal, or you directly write it into Drupal’s editor?

I usually write it in Word first, because I find it easier to edit it. I write the article there and send it to the scientists and they tell me if it’s correct, and after it’s edited I copy it into Drupal. I also add photos from the photo archives.

So you mentioned WordPress. Do you have some experience with it? Do the others in the team have some experience with it?

I have my own personal blog which is a bit sleepy at the moment. It is in WordPress so it’s a very simple blog.

So it was less of a steep learning curve for Drupal.

Yeah, I’ve also used Escenic and Vortex so I tried several publishing systems. The editor part is almost the same in everyone. So if you learn one it’s easy to learn a new one.

What do you know about taxonomies and tags?

That was something that came up when talking with others in the communication section and with the dean. What kind of tags we want to show, because it was important that they had to be stable, you had to choose one of them. Because with free tags we started out with the writers, and then we had to see which ones we would keep. But we had many discussions to find each tag we wanted to have there. It involved the whole faculty actually.

What do you know about Semantic Enrichment?
I don’t know much about it. I understand it’s about a more intelligent web, and search optimization. When I think about search optimization it’s like, oh, I should use this word ten times in the article and then it will be shown high up. But when you are a journalist and you use the same word over and over again, the article is going to be bad, it is not good storytelling. So I hope this project will make more visible the site. I’m actually very curious about what it will do.

Have you received training in any of the relevant technologies?

Yeah, I’ve had a short course in Google Analytics so I know the basics. And we also had a short course in Mailchimp, and that’s what we are working on now. Gisle had to show us how to use the site of course. Two years ago I had a course on social media, but we need more help to be visible.

Do you generally prefer having restricted options when you interact with an interface so that you can avoid mistakes and errors?

Now we have this standard tags that are visible and also the free tags are also visible. I think that’s ok to do it this way. Now I can say to people ‘If you’re interested in the universe you can go to the universe and get all the articles that have been tagged about the universe’. Gemini for instance, they have undertags. So that may happen that we want that, but we haven’t talked through it yet. It can happen that it will be in the next version of the site.

Do you have any difficulties with the editorial interface, is there something that confuses you or something that you’d like to see done differently?

No, it’s more about small things with the design I guess. Like that big drop down area in the top. There is some mess in the comment section. But we will take this one at a time. And I don’t know if you’ve seen the mobile version. It’s small things. Like, making a fact box is not very user friendly, but Gisle made it in a way that we can do it. It works quite well for being such a new site. We take that little by little.

Are there any comments about the WYSIWYG editor?

I think we need that, because if someone does something wrong it’s so much hassle to fix it. So, a lot of people are writing blogs and articles and there
will be more in the future, so from a user perspective it has to be user friendly.

How do you feel about learning and using yet another thing?

If it’s easy to do it and it doesn’t take much time, yeah. So if it will make our site more visible, I’ll do anything.

What’s your take on reading help pages, tutorials, and manuals? Do you prefer getting help along the way?

I think it works well the way it is now, with brukerhåndvisning. And there is a different team for bloggers, for writers. I think that’s an ok way to do it. For example if I want to publish a video I just look for it and it’s easy to find it there.

How do you see the timeline of the project evolving?

I am not a very patient person, so I think it has taken too much time already. But, I think we now are in a place that we have a good organization to make an article and publish an article every day and we are working now to spread the news more. Having the newsletter up and running. And then we’re going to hire someone to fix all the small things, and also working on the next version of the site. A lot to do with the undertags and that kind of thing.

Would you consider getting more people in? Writers, bloggers?

Yeah hopefully but that will also take some time, because we hope we can double it in a year I guess. When it comes to writing articles I hope that we’ll get more funding for that.

Interview with Communication Responsible (Ida Marie Bjørknes)

The interview took place in one of the colloquium rooms in IFI at 12.02.2016

Tell me a bit about you duties in the university, what the typical working day is like, if Titan is a full time employment for you, and if not how it fits in your day.

I am an editor in the faculty, and I am responsible for the social media, the Facebook channel and Twitter. And in addition I have the responsibility for internal
communication within the faculty – the employees. And then Titan is in addition to that.

How much time do you find that you dedicate to Titan?

It kind of depends which phase of the project we’re in. Right now I would say 30% of my day.

What are you responsible for when it comes to the website?

I am in the project group, with Gunhild and Gisle. I am working with them to keep the site running. From the start of the project and so now, so we are trying to develop Titan further, trying to get the newsletter and Google Analytics and look at which ways we can improve the website. So the writing and stuff, I’m not doing that.

In a Titan working day, what is the full timeline, how do you start, how do you finish?

Well I usually go back and forth between Titan and other tasks. I always have a plan but something comes up in the day or the week that I have to finish before. So it kind of goes off and on again.

The website has a set of roles. What is your role, how it relates to the others? Do you perceive any hierarchy?

I am an editor, but I’m also in the side of all that because I am just working with the website, trying to improve it. Not writing articles, proofreading and that kind of thing. I do it sometimes if I’m not really busy. And Gunhild is the project leader. I don’t have leadership.

So how are you working? Are you reviewing the site, and then keeping notes, and then meeting the others to discuss?

Yeah. And then Gunhild and I are trying to develop the newsletter so we’ve been doing that a lot and we’ve taken a course on how to use Mailchimp, and we’ve been communicating with a firm which will help us to time the newsletter properly. We tried to do it ourselves but it’s complicated. So we’re getting help now doing that.

Have you taken any other courses?

No I think that’s the only one we’re taking. And I did a course in Google Analytics on Friday. We’re also using that on an additional website, uio.no.

How long have you been involved in the project?

I was along since the beginning. I just started in this job November 2014, and December 2014 they started talking about Titan and I got into the project group. Gunhild started talking about Titan before I got here.
So what is the purpose of Titan?

We try to let the world know about all that fantastic interesting research that is done in the faculty, because there’s a lot of amazing research and it’s so hidden, so we cannot learn about it if the researchers are locked in their offices doing their thing. We have to let society know about them. And then when society gets to know about all that amazing research we do, then we will get connected with politicians and research funds and maybe recruit students and be associated with really good research and good students.

Would you modify the way you present your content according the results of the analytics? Would the analytics have an effect?

It might have. We want to know who the readers are and try to meet them in a way but also try to make others interested as well. But we try to reach abroad to our readers. But once we know who our readers are we might tweak the content. It is a possibility.

You are following a popular science approach.

Yeah, we’re trying to reach the man on the street – so that people from all backgrounds can really understand what we’re into and it’s kind of difficult sometimes. I don’t have a background in science.

What is your background?

It is in communication and media. I have a master’s degree in media, communication and governance.

What have you learned from your involvement in Titan?

A lot, both when it comes to bureaucratic stuff and risk analysis, so that it doesn’t go against the university’s strategic guidelines, and also how to make a page. A project is much more complex when you’re in it rather that what you thought before you started.

What do you know about Semantic Enrichment?

I guess it’s about the language and the words you use on the website, and tags and how to use that to get better results on google.

Did you have any expertise in other relevant technologies to Drupal. Did that expertise help you?

I’ve been working with similar programs before, like SharePoint and WordPress a couple of times. And there are some similarities between them. But also differences, like Drupal doesn’t have a map tree, it’s almost flat, more based on searching in the background, like the images in the folders. There is always something new.
A big subject of discussion was the WYSIWYG editor. What is your take on it?

Yeah I’ve used the editor. I’ve used the other one, the one that Gisle likes but it’s difficult only if there is some error on the WYSIWYG editor, like invisible error. So I go back to the other one in the background. But I think the WYSIWYG is helpful for people who are not tech savvy, because it reminds them of the word program, like, b is for bold. I think it’s not that dangerous. Maybe they are afraid of deleting a page.

Whenever you’ve used this editor, have you first typed into word and then pasted in the editor, or do you directly type it in the editor.

I usually paste it in because I find it easier to see the article; the window is so small, so narrow. It’s easier to see the bigger picture of it all, the overview.

Taxonomies, have you being along on the design?

Gisle has been working on that. We have given him input on which tags we would like to do. The Titan group asked for tags but Gisle puts them in Drupal.

What are the groups?

The project group is Gisle, Gunhild and myself, and there’s the news desk group, that’s Gunhild and journalists called Espen and Johannes, and in addition to those there are some people from some of the institutes who are also writing articles and they join whenever they have time. And the bloggers are kind of on the side. Gunhild is in charge of all. Both project and news desk, and in dialogue with the bloggers. And that’s where I also step in sometimes.

Do you prefer having restricted options in order to avoid errors, or having them all laid out for you?

I prefer restricted especially when it comes to those who put their articles into Drupal, because they don’t use it as much so they can sometimes forget, so my opinion is that erase all channels to do errors, or misunderstandings. I think some people are more motivated when the challenge to do errors is minimized. So, it makes it easier for them to do the job and they feel safer. I personally would like to have all the options.

When it comes to tutorials, help pages and manuals, do you actually read them?

I do read the help descriptions that Gisle wrote. It explains how to embed a video in the article how to attach tags. It’s really into the point of what we need to know, and then I can google things, in points, like one two three. And if it’s like a long text then that’s not good.
Would you prefer it if there was help along the way? Instead of reading a help page to have a popup explaining what to do at that point, or would that be too intrusive?

I don’t like windows popping up in webpages. I like to do it on my own especially when it’s so easy to find Gisle’s help descriptions.

What about having a questionmark or an exclamation point that you would click and then get help.

That one I like, but then you have to click yourself and read it. I’m for the buttons you can push.

How do you feel about having to learn and use yet another thing. Is it something that people would use long term if they saw the benefits? Or something that they would use for a couple of months and then abandon?

I think that if they see the benefits then they will do. I’m kind of used to learning new stuff cause I haven’t been here for so long, a year and a half almost so there is a lot of new stuff so I don’t really mind.

Have you used any of the inhouse systems that the university has?

Yeah I’ve used vortex.

Do they change things often or is it stable?

It’s stable and they let us know if they change anything, so for example on Thursday they are going to do something in vortex so there is a message on the backside of vortex that there will be maintenance work at around 9 o clock. So don’t be inside the program. And afterwards they always write what they did, It’s quite nice.

Do you have any difficulties currently, anything you’d like to see change?

There’s this small thing, when you put up a picture in the article box then everything outside the box, like title, get so small like size 8 maybe. But the important thing is that it looks good for the readers and then we’ll deal with the backside stuff. We’re planning to do some improvements in the front end design, like the banner on top it’s quite big. So we might minimize it or remove it, or have it when you scroll up again instead of down. And then we might minimize some of the icons and the carousel. It will be interesting to see how that works. Then we also have to work on the responsive design in iPads and mobile phones.

How do you see the timeline of the project?
Right now the plan is to do some Facebook campaigns, and then we have to work with Mailchimp, and we hope to receive the first draft next week and then we have a list of alumni, and we will contact them and let them know that we’ve done that and tell them to subscribe to our newsletter if they want to, and then plan for the version 2.0. We’re also trying to get new bloggers. The goal is to have one new article every day. And then there’s just maintenance and small tasks, and people coming asking how to do this and that in Drupal.
Appendix B: Thinking Aloud Sessions

Informant 1 (Editor)

Informant 1 begun by filling in the title and body of the new Article using the WYSIWYG editor, taking about five minutes to type in original content and making the process as realistic as possible. In the meantime she explained how she fills in the various fields. She asked if she should insert a picture and I prompted her to do so, as prepopulating fields with image data is a feature of the module. She proceeded with publishing the article without paying notice to the module.

‘So I guess that’s it’.

I prompted her to inspect the publishing options area, as displayed in the following screenshot, and said:

‘What should I do here?’

![Figure 10](image-url) The module as it appears in the publication options area, seen from the administrator interface

I prompted her to click on the vertical tab and inspect the different options. Once she did, she was presented with the different meta tag categories which are collapsed by default.
She said: ‘Now I don’t know what I should do here’.

I instructed her to click on any of the boxes and see how some values are prefilled. When she noticed that the description field was empty (because the Sammendrag field was empty too) she said: ‘Is that the Ingress?’

I briefly explained then that the idea is to inspect the prefilled values in case she wishes to change some of them, and then to ‘fill in the blanks’. I used the Standout field as an example and explained that if there is a source that broke the story, she could get the URL and paste that value in the field. I then realized that she did not read the field descriptions and that it was unclear what type of value the field expected, because I only stated this in the end of the description. I explained that it is not obligatory to fill all of the fields. She then questioned: ‘What is the use of the image?’

I explained that it is there for presentation purposes in various social networks. I then explained what is the purpose of each meta tag categories.

She said once more: ‘I’m not sure I understand, for example this Google news keywords... If I fill this in the article will rate higher’

I explained a bit more thoroughly what was written in the field description. She went back on the standout field and commented that: ‘Sometimes I cannot know who broke the story’.

I replied that it is not necessary to fill the field in, and noticed that she referred to URLs as ‘links’ which made me consider to change my naming policy. She then wondered about the difference between google standout and original source, and said: ‘If they are similar just keep one of them because we are trying to teach other writers too and it should be as simple as possible’.

She then wondered:
'Is it possible to fill this both for editors and for writers?'

I explained that prefills are available for everyone and that if she believes that there should be restrictions, I could remove privileges from some of the roles. She responded with:

‘Yes, I guess I have to think about that’.

She then wondered about what the ‘See Also’ tag is about. I explained more thoroughly what was stated in the description and she said:

‘Can we put articles that are not ours?’

She then inspected the Open Graph Image URL and Secure Image URL and asked me what the difference was. I explained and she replied

‘Why is it two there if I don’t have to fill it?’

She then said:

‘So this is about Facebook... Does this mean that it will be automatically posted on Facebook? Because now we go and post it there’.

I explained that the article shows up on Facebook only when they post it there themselves and that the article will not show on Facebook when she clicks ‘Lagre’.

I then explained about the article type choices, and whether she would like the field to be prefilled with the first value out of the standard category field or if it should remain empty, leaving the choice to the journalist. She said:

‘The journalist should choose.”

and I later realized that she thought that the prefilled values cannot change.

Upon inspecting the Twitter Cards category she wondered about which fields expect personal Twitter account. Once I explained she replied:

‘So if someone from the editorial team wants to put their account... will it show up on their accounts or...?’.

She also wondered:

‘If I have a personal webpage can I put it on the Contributor?’.

She opened the Dublin Core Advanced category and said:

‘I’m not sure what that is, is it.... No I’m not sure I understand what this is.’

She directly moved to Google+ and said:

‘We don’t have a Google+ should we get one? Because I don’t know if we have the time for that.’

I explained less thoroughly what was written on the description and focused on Rich Snippets and Google Cards, and then I opened a new window googling ‘Microsoft’ to exemplify what I was talking about. I focused on how these previews may make the content more noticeable.
She then wondered about the order in which the fields are prefilled and I explained that once she fills the standard fields the meta tag fields will be filled accordingly and not the other way round.

She then wondered:

‘About choosing the keywords, is it the most important thing Google is looking for?’

At about 35 minutes she was done with editing and I explained that she can click ‘Lagre’ and everything would look as usual, and that the process is the same for the other content types. She was apprehensive to save the changes

‘is this thing going to be published now?’

and I told her it is not necessary.

I then opened an existing article and told her that the process is similar. This time she started filling in some keywords and when offered the choice to have prefills she said:

‘Yeah yeah that’s good, but if I want I can go in and put more words. It can be filled in automatically and then you can go in and change it.’

I stated that all the prefilled values can be changed. She then kept inspecting and filling in fields, and returned to the Image and Secure Image fields she said

‘As few as possible’

At that time I did not provide Prepopulating for image width and height. And she said

‘I can look into the archive’.

She then left the article editing page and mentioned that she would be interested in an automatic saving functionality. She returned to the article, tried to fill in more fields, and clicked ‘Lagre’. No errors came up and I informed her that whenever errors occur, they will appear in the same place where all warnings appear and that the field that caused the error will be highlighted as normally.

She requested that more fields are prefilled, and asked once again about the allowed values in some of the fields.

We then had a more informal discussion where she mentioned:

‘Everything that’s easy and quick to do is nice, because we have so much else to do’.

When asked about the vertical tabs, which are displayed in the figure below, she replied:

‘No I don’t need this, I remember I need to fill in all this’. 
Informant 2 (Editor)

Informant 2 started creating a new article by copying an existing article. She filled in most of the information and located the module instantly. She read the descriptions for a while and wondered:

“What kind of keywords are good for Google news?”

After I explained she asked:

“What does canonical mean?”

While inspecting the Google Standout and Original source articles she mentions:

‘If they are similar, it would be better to keep up because we don’t want to mess up’

She then applied some changes to the article’s fields and noticed the corresponding changes to the meta tag fields. Upon inspecting the Open Graph Image and Secure Image fields she proposed:

‘Rotating the fields depending on which one should be filled.’

She then wondered which dimensions she should enter and suggested the possibility:

‘To extend this one (scald) to show the dimensions.’

Afterwards she questioned whether the locale refers to the language of the values of the tags or the meta tags themselves.

She also said that she likes prefilled fields and prefers the date being prepopulated with text rather than having a calendar widget, a statement
that came after my prompt. She then mentioned that the publication and modification date and time were confusing because the date of creation and the date of publication of an article are different.

She then inspected the different twitter card types and wondered which one was appropriate for the case.

She mentioned that:
‘Things should be as simple as possible because of the time pressure.’

She also wondered if Titan should have a Google+ account. And made various recommendations like:
‘Change the Article author label into Author’s Facebook Page’
‘Keep the boxes collapsed’
‘Skip the code if no one understands it’
‘Rename Open Graph into Facebook’
‘Maybe gather all the URLs together’.

At that point we discussed about a hybrid approach where the categories remained as they are but the fields that expect URLs are grouped together instead of being alphabetically sorted by meta tag name as they currently are.

It took her 28 minutes to create the article and 10 minutes to edit the existing article and add meta tags that were missing. We then continued our discussion during which she mentioned that she likes that there is uniformity considering Drupal’s interface, and that the Date fields where the most confusing as it was unclear whether the publishing time will be affected. She also recommended that the basic and advanced meta tags are put together as this is
‘Less intimidating and we want people to fill in as much as possible, and to do so they need to open the boxes’.

Finally she suggested that I rename all the categories.

**Informant 3 (Author)**

Informant 3 started the session by inspecting the article creation interface and said:
‘This is very recognizable, so I should start by writing my name here I guess.. two choices came on, I’ve always been wondering about that. All writers should do it in the same way’.

I explained that the first choice, First name Last name, is preferable in comparison to the second, First name Last name/UiO.
‘I agree with what you said, people already know that this is the University of Oslo. So that’s what I generally use, so maybe you could take away the other choice’.
He kept inspecting the standard fields and said:

‘Ingress is also very familiar’

I explain it’s the same interface. He added an image and a caption.

‘This tagging is a matter of internal discussion in my head. What is the smartest way to tag an article? It should be in connection with the feed categories, but I usually write something... I try not to write pharmacy as a tag, because that’s already a choice, so I can write something else that is describing the article. But this is also difficult sometimes, to find the best way to do it’

By that point he finished filling in the standard fields, he started inspecting the module:

‘So now we are going into new territory... What I should do here i guess is open up these...Grunnleggende meta tagger, I don’t know what that is...I have to read this in order to understand it...Okay, so I’m not supposed to do a lot there’.

I explained that he doesn’t need to do anything unless he wants to change something.

‘This Alle rettigheter forbeholdt could change but now I see no reason why’. When editing the Facebook meta tag set he commended:

‘Innholdstype.. I could choose something else but I don’t see any other choices that fit, I could put university but no... If I don’t understand it I won’t change it. URL looks good, URL til relatert innhold...I could enter some other URL which will be complimentary to the article I guess. I have nothing in mind in this context but I understand what you mean. URL-adresse til bildet.. That looks good, Type of picture, I don’t see a reason to change it. Lokalisering... I think that should be språkform or something. Forfatterens Facebook-profil, is that me as a person or Titan? I know what to do, I have a Facebook account and I can enter the address there. That’s easy, I don’t have to do it right now. Kategori, I already entered that didn’t I?’

I explained the two different options; having the field prepopulated with the first choice or empty, since only one value is required and he replied:

‘The ordering is not my choice there so... but I can choose anything I like, so that’s alright then’.

And then continued with:

‘Date and time...that’s alright’

Adding Twitter meta tags followed:

‘I use Twitter but I’m not very familiar with it. I’m not sure I understand what a Twitter card is. Now I immediately understand it when I look at the choices. This has a big picture and medieauspilling will be if I enter a video.'
So I prefer…. large image. Forfatterens twitter konto.. That’s easy to understand, that’s me again. URL is no problem, nothing difficult here’
‘Dublin Core, I don’t know what that is. Is that something secret in Google or..?’

Once I explained what was written in the description he commented:
‘It’s perfectly understandable’

He continued adding and reviewing the fields:
‘I will choose Naturevitenskap…and the next ones are filled in automatically. URL adresse til bidragsyteren? That person, is that me again?’

Once I explained in greater detail what was written in the description he replied:

‘If I interview a scientist who has been financed from the research council of norway then I could put that. They have a very easy to remember address, or do I need to fill out the whole address? Date is no problem, type is text, filformat ..html. I don't see why I should change that..Kilden.. If I write an article there is no other source than myself so I guess I shouldn’t write anything there. Relevant innhold, but is that going to be entered into the article in a way that readers see it?’

At that point I explained that none of the meta tags are visible by the users.
‘Okay the next chapter is about google. I don't see any problems, most of it is entered automatically… Utgiver…hm.. We don’t want to own the content we want to give it away. I should leave this blank’

Creating and saving the Article took about 16 minutes. I urged him to create a new article to see if he would notice Google News being unlocked as the criterion was now satisfied. He then remarked:
‘It’s quite slow today’

I explained that my module is slowing down the site while he started to create a new article.
‘This keyboard is not exactly like the one I work with. Should anything be different now? It looks like what we did… It’s the same isn’t it? Google News! Agh, that’s new. To choose nøkkelord in Google, that’s important, because this about getting higher up in Google Search. Is it possible to see a list of keywords that are sort of, more popular and match them with the content in our article?’

I told him that such a feature can be implemented, but for now it’s free text, and he resumed:
‘Is that written by somebody else? Ok, so it can be one of my own articles. I guess most of what I’m doing is not written by anybody else.. I get what you mean.’

He then started adding Twitter Card meta tags:
I could change the descriptions, it’s not a bad idea, because this is going to be visible in twitter. It’s a good idea to think creatively about this. That’s a good one it’s an important one. Yeah, there’s nothing more I need to do’
‘To Google then. Then I can add a new description. Which picture is that, I lost track of it’
I explained and he replied
‘So if I entered a picture that one would be filled’
I confirmed this, and then he saved the article. An error occurred in one of the fields and he suggested that the error description links to the field so that he doesn’t have to scroll down.

Informant 4 (Blogger)

I started the session by explaining the purpose of the module. The informant wondered:
‘But do I have to write content?’
I confirmed and explained about prepopulating. She started creating content, and once she was done, she opened the module and said:
‘Browse available tokens...what is that... I’m a little unsure about this’
I realized that the token browser should have been removed and informed her about it. She closed it and before filling in the related content meta tag she said:
‘Can I search?’
She proceeded with entering a URL in the field and when she moved to the Image URL field commented:
‘This is a little bit confusing.. The URL for the photo? What photo?’
‘It’s actually nice that I get to see all this. Maybe it will lower the threshold to publish something now. Google plus, that’s nice. Creative Work, that’s interesting... It sounds a bit better than a blog post’
I then explained about the nature of the term and how it is the most general of them all. By that time she has finished filling in all the meta tag categories.
‘Yeah, that’s as far as I can go. Should I save it?’
Some errors occurred in the fields expecting URLs, and she clicked the back button, so all the data got reset. She commented:
‘Oh no, did I ruin something now??... Anyway... It’s nice to publish to Google+’
I explained that the content is not automatically published and she replied:
‘Yeah I get that. Of course it doesn’t why would it, I’m not logged in with my Google+ account. When it says Google+ you just immediately think that, but
yeah. It’s nice to have the opportunity for some extra outreach—which I suppose is the purpose. People should be quite good at using tags right?”

By that time I decided not to test the Gamification part, and asked for feedback. She pointed at the module’s description and said:

‘This is a nice explanation’

We then had a short discussion about the purpose of adding meta tags and she wondered about how content gets posted on social media. I explained that it doesn’t happen automatically during publication. She then commented on her input during the session:

‘I guess it’s worth it.. I mean, everyone will start using Titan and they need to onboard a lot of new bloggers, obviously this interface will be completely unknown to them so I guess it’s nice to also have someone who hasn’t really posted before’

She then added:

‘I wondered about Dublin Core, I don’t know what that is’

I explained what was written in the description and she replied:

‘This is fine but I would maybe...rather say why they are used and where they are used. Otherwise I would have left that blank because I didn’t really understand what that would do for me. I think it’s nice, I like the opportunity of having your post more visible’

Informant 5 (Author)

The informant started the session by creating a realistic article. She mentioned that what she was typing in was an idea she had for a while and that whenever she posts on Titan she copies and pastes from Word. She then acknowledged:

‘I have a Mac so I scroll the other way round. If I take too long, this is why’

After filling in most of the standard fields, she asked:

‘But should I use a toppbilde?’

I encouraged her to do so. And then she said:

‘And now I just save it?’

I replied negatively, as she had not used my module, but she had already clicked Save. I suggested that she could create a new article after I delete the old one, being an administrator. I then showed her where the module can be found and explained about field prepopulating. She started creating realistic content once more. When she completed filling in the standard fields she questioned:

‘So what do I do now?’

I urged her to click on the module and inspect the different categories. She clicked on the Sidetittel meta tag and said:
‘So um… Using a comma I can add keywords for Google search’
I explained that she can always add keywords and descriptions as long as it is relevant for the field in question. She then realized:
‘Agh of course I don’t want that’
Upon inspecting the description field she commented
‘This is more what the article is about. I suppose this is a good thing because very often the things you add as a site title or undertitle are not descriptive. They are sort of just to make someone click or make it more interesting’
She started changing the Beskrivelse field and continued with inspecting the other categories:
‘So this is supposed to make it easier to define for Facebook, and this is the same URL as up here, but this descriptions is what will be shown on Facebook, because now you have the undertitle. You could have a button like “Do you want the description to be the undertitle?” And then you click and it’s there. Oh can you add more URLs that are related to the content?’
I explained that only one value is required.
‘So here I could add my Facebook profile? It should be my Facebook URL, not my name. So there is a difference on who is issuing the content and who is writing the content. Will people be able to click the article and go to my Facebook page? And will it be possible to add a time where it will be published here? Can you also put a field on when it will be published on Facebook or do you have to go to Facebook to do that?’
I shortly explained about the scope of the module, and a few moments later she said
‘And the same applies for Twitter, right?’
When she was done with Twitter and collapsed the category she said.
‘What’s Dublin Core?’
I retold what was written in the description and she started filling in the form fields. A while later she wondered:
‘So can I use for instance my personal page at UiO? And would that be understood as a person or a company’
I replied that it will be understood as an Agent, an entity more general than a person or a company.
‘But usually when it’s written on Titan it’s an original page’
Once she was finished with all the categories she said:
‘So I save’
Two errors occurred as the Author’s Facebook profile field was filled in with her first and last name, and the Contributor field had an incomplete URL.
‘Should I…I can find my Facebook, should I sign in to do that?’
I told her it is not necessary if she remembers the postfix of her Facebook profile, but she couldn’t recall it, so instead she typed what she believed was correct. Once she fixed the error and saved the article was created successfully and I urged her to create a new article. Once again she put effort into generating realistic content. However she did not notice the Gamification notification. She noticed the Google News category and exclaimed:

‘Mm this one’s new. So yeah. This is interesting. I can write in Norwegian... It doesn’t matter.’

She read the descriptions and said:

‘I should probably be very specific. Now I used what I typically use in an article, but if I actually wrote this I would be more specific. So here I can add links?’

I explained that this is the case, but only if the criteria for quality journalism are satisfied. She decided to skip the rest and save, successfully creating the second article. When I asked for feedback she concluded:

‘This is interesting; this is good for me if I get more views’.
Appendix C: Forms of informed consent

Request for participation in research project “Semantic Enrichment for non-expert users in a News Outlet Website”

Background and Purpose

This project is carried out in the context of a master’s thesis at the University of Oslo. The purpose is enabling Semantic Enrichment for non-expert users in a news outlet website powered by Drupal, following the User Centered design paradigm. Informants are selected based on their role and involvement in the target website. Diversity regarding the tasks they perform and the time they spent on them is a main criterion for selection.

What does participation in the project imply?

Participating in the project entails taking part in an interview, workshop, and observation, none of which will last longer than 45 minutes. Questions will concern the informant’s role and involvement in the website, as well as general questions about the website and personal preferences regarding interface elements. When applicable, audio will be recorded using a digital recorder, and notes will be kept simultaneously.

What will happen to the information about you?

All personal data will be treated confidentially. Access to the results of the research will be provided to anyone with access to the thesis. However, all recorded material will be destroyed, and all notes kept will be anonymized when the project is terminated. The informants will be represented by the respective roles in the notes. The project is scheduled for completion by August 2016.

Voluntary participation

It is voluntary to participate in the project, and you can at any time choose to withdraw your consent without stating any reason. If you decide to withdraw, all your personal data will be made anonymous. If you would like to participate or if you have any questions concerning the project, please contact Magdalini Fotiadou, 40577569, magdalif@ifi.uio.no.

I have received information about the project and am willing to participate

(Signed by participant, date)
Request for participation in research project II “Semantic Enrichment for non-expert users in a News Outlet Website”

Background and Purpose

This project is carried out in the context of a master’s thesis at the University of Oslo. The purpose is enabling Semantic Enrichment for non-expert users in a news outlet website powered by Drupal, following the User Centered design paradigm. Informants are selected based on their role and involvement in the target website. Diversity regarding the tasks they perform and the time they spent on them is a main criterion for selection.

What does participation in the project imply?

Participating in the second phase of this project entails performing some tasks while ‘thinking aloud’ - where the informants will vocalize their thoughts while executing the tasks. Audio will be recording using a digital recorder, and notes will be kept simultaneously. Task execution will be timed. After the ‘thinking aloud’ session is completed, the informant will be asked questions regarding the features of the module, and will be prompted to give general feedback and suggestions.

What will happen to the information about you?

All personal data will be treated confidentially. The informants will receive transcripts of all the digital recordings. Access to the results of the research will be provided to anyone with access to the thesis. However, all recorded material will be destroyed, and all notes kept will be anonymized when the project is terminated. The informants will be represented by the respective roles in the notes. The project is scheduled for completion by August 2016.

Voluntary participation

It is voluntary to participate in the project, and you can at any time choose to withdraw your consent without stating any reason. If you decide to withdraw, all your personal data will be made anonymous.

For any questions concerning the project, please contact Magdalini Fotiadou, 40577569, magdalif@ifi.uio.no.

Consent for participation in the study

I have received information about the project and am willing to participate

(Signed by participant, date)
Appendix D: Coding Standards

Drupal’s Form API

The Form API allows generating, validating and processing HTML forms. Some of the implications of the Form API are that rather than outputting HTML, the developer creates an array and lets the engine generate the HTML. Since the form is represented as structured data, adding, deleting, reordering and changing forms is easy. Moreover, additional form validation or processing can be added to any form.

Below some central stages of dealing with forms are described (Tomlinson & VanDyke, 2010, pp. 239-264):

- Initialization: Three variables are important:
  - $form_id, contains a string identifying the form
  - $form, is a structured array describing the form
  - $form_state, contains information about the form, such as the form’s values and what should happen when form processing is finished.

- Theming: the most flexible way to theme forms is to use a theme function for that form, or element. This is done through implementing hook_theme(). Then a template_preprocess function gathers all the elements form the form and makes them available so that the themer can control how the elements appear in the form. If there is no function matching the “theme_form_ID_name” pattern, a #theme property can be also specified.

- Validation: The purpose of validation is to check that the values submitted are reasonable. If validation fails at any point the form will be redisplayed with the validation error shown to the user. The form_set_error($element[‘title’], $msg); call may be useful here. A validation function for a form can be assigned by setting the #validate property with $form[‘#validate’][] = ‘foo_custom_validate’;.
  Additionally element specific validation is possible, if there is an #element_validate property defined for an individual form element. This is how validation was added in the context of this thesis. The custom validation functions are passed the $form_state and $element variables.

- Submission: in a similar manner, the function that handles form submission can be assigned by setting the #submit property with $form[‘#submit’][] = ‘foo_custom_submit’;.
- Fieldsets: splitting the form in different fieldsets is easy with the Form API. Each fieldset is defined in the data structure, and has fields defined as children.

- Changing a form is done through implementing `hook_form_alter()`.

- Properties and form elements: `#description`, `#required`, `#attributes`, `#type`, `#attached`, `#default_value`, `#element_validate`, `#title` and `#weight` are some of the properties used in the forms I manipulated. Built-in Drupal form elements I used include texfields, textareas, select elements and check boxes. An example of a form element follows:

  ```php
  $info['tags']['og:image:type'] = array(
      'label' => t('Image type'),
      'description' => t('The format of the image'),
      'form' => array(
        '#type' => 'select',
        '#options' => _metatitan_opengraph_image_type_mime(),
        '#empty_option' => t('- None -'),
      ),
      'weight' => ++$weight,
  ) + $og_defaults;
  ```

**Drupal Development Best Practices**

As mentioned, Drupal code adheres to strict standards. Below are some of the best practices I tried to adhere to:

- Line indentation and whitespaces: two spaces were used for indentation instead of tabs, while `\n` was the end of line character.
- Operators have a space before and after, apart from unary operators.
- Spaces are placed between the type and the variable in a cast.
- Control structures place parentheses with spaces in between, while curly brackets are always used.
- Function calls use single spaces around the operator and no spaces between the name of the function and the parenthesis. There is no space between the parenthesis and the parameters.
- Function names are written in lowercase and based on the name of the module they are a part of. Underscores are used instead of camel case.
- Array elements are placed on their own lines for readability and maintainability. A comma is placed in the last array element as a measure of caution.
- All documentation blocks use the doxygen comment style guidelines for the JavaDoc style, which consist of a C-style comment block
starting with two *'s. Constructs like `@param`, `@return`, `@file` and `@see` are used.

- `<?php ?>` notation is used instead of the shorthand notation.

jQuery Best Practices and Design Patterns

The following design patterns and good practices were followed:

- **DRY Principle**\(^{26}\): whenever possible, redundancy was avoided.
- **Single var pattern**: This pattern instructs using a single `var` statement at the top of the function to declare multiple variables delimited by commas. Initializing whenever possible is also a good practice to avoid logical errors. The pattern provides a single place to look for all the local variables prevents logical errors that might occur due to hoisting, helps minimizing globals and results to less code.
- In order to improve maintainability and predictability, Drupal’s naming and indentation conventions for JavaScript were followed.
- **Literal expressions** were used instead of the `new` operator\(^{27}\) when creating arrays and objects.
- Instead of adding new HTML elements to the DOM with `document.createElement()` the jQuery equivalent was used.
- **All JavaScript code** is declared inside a closure wrapping the whole file, and this closure is in strict mode\(^{28}\):

  ```javascript
  /**
   * @file
   */
  (function () {
      "use strict";
      // All the JavaScript for this file.
  }());
  ```

  Using strict mode means that undeclared variables will halt the script. All variables are declared once and the declaration happens in the beginning of the function.

- **Functions** are objects that can be created dynamically at runtime, they can be assigned to variables, have their references copied to other variables, can be augmented, passed as arguments, and returned, and can have their own properties and methods. Functions are first-class objects and they provide scope. There may be named function expressions, or unnamed function expressions – also known as

\(^{26}\) [https://learn.jquery.com/code-organization/dont-repeat-yourself/](https://learn.jquery.com/code-organization/dont-repeat-yourself/)

\(^{27}\) [https://www.drupal.org/node/2297057](https://www.drupal.org/node/2297057)

\(^{28}\) [https://www.drupal.org/node/172169](https://www.drupal.org/node/172169)
anonymous functions. Functions can be declared in a similar way to other languages, or they can be immediate – executed as soon as they are defined. The JavaScript Module Pattern\(^{29}\) was used to organize the code. First, it has the advantages of privacy, because functions create scope. It allows for maintaining the private variables’ values, keeping only the module – function name available and leaking nothing else in the global scope, as well as promoting maintainability and reuse. All the functionality is contained in an IIFE (immediately invoked function expression) which returns an object accessed inside `Drupal.Behaviors`.

```javascript
var myJavaScriptModule = (function(){
    var myJS = {};
    myJS.moduleMethod = function () {
        // implement functionality
    };
    return myJS;
})();

Drupal.behaviors.myModule = {
    attach: function (context, settings) {
        // Use the JavaScript Module.
        myJavaScriptModule.moduleMethod();
    }
};
```

- DOM access is the most common performance bottleneck (Stefanov, 2010, p. 183). This means that whenever possible, accessing the DOM in loops was avoided, DOM references were assigned to local variables, and the length was cached when iterating over HTML collections. Collections are live queries against the underlying document, so every time a collection’s length is accessed, the DOM is queried live. Hence, a pattern for `for` loops is to cache the length once and use it throughout the loop. Caching the length when iterating over collections is anywhere between two and 190 times faster (Stefanov, 2010).

```javascript
for (var i = 0, max = array.length; i < max; i++) {
    // do something with array[i]
}
```

The single var pattern may also be followed in this case resulting in:

```javascript
var i = 0,
    max,
    array = [];
```

\(^{29}\) [http://atendesigngroup.com/blog/requirejs-and-javascript-architecture-drupal](http://atendesigngroup.com/blog/requirejs-and-javascript-architecture-drupal)
Appendix E: The final, redesigned module

Figure 13 The collapsed module as it appears to an editor, before Gamification elements are added.

![Gamification status message]

Figure 14 Gamification status message
Legg til meta-tagger for å øke synligheten og optimalisere hvordan innholdet ser ut når det deles på sosiale medier.

**GRUNNLÆGGENDE META-TAGGER FOR SØKEMOTORER**

**GOOGLE NEWS**

**FACEBOOK**

**TWITTER**

**INTERNASJONALE STANDARDER**

**GOOGLE**

Figure 15 Collapsed module interface after Gamification conditions apply

Legg til meta-tagger for å øke synligheten og optimalisere hvordan innholdet ser ut når det deles på sosiale medier. Innholdet blir ikke delt automatisk på sosiale medier.

**GRUNNLÆGGENDE META-TAGGER FOR SØKEMOTORER**

Disse metatagene kan brukes til arkivering i søkemotorer, for å vise biter av innholdet i søkeresultatene, eller vise informasjon om siden på brukerens nettleser. Meta-tagene definerer også URL-adressen til innholdet, et relevant bilde, og rettighetene over innholdet.

**Sidetittel**

Lingvistikk, statistikk og franske hatter

Teksten som skal vises på tittelinjen til brukerens nettleser. Denne meta-taggen brukes også som tittel når brukeren lager siden i bokmerker eller favoritter. Feltet er forhåndsgyldig, men kan endres hvis ønskelig.

**Beskrivelse**

Du har kansje hørt at datamaskiner kan gi beregninger basert på 0 og 1. Med språkteknologi, kan datamaskiner bruke 0 og 1 til å forstå menneskespråk.

Et sammenlign av sidens innhold, på inntil 150 tegn. Denne meta-taggen kan brukes av søkemotorer for å vise et utdrag av siden i søkeresultatene. Søk som er beskrevet er unik, ellers lar feltet stå tomt. Lær mer om beskrivelsen i denne videoen fra Google6. Feltet er forhåndsgyldig, men kan endres hvis ønskelig.

**URL-adressen til bildet**

https://titan-app-test01.uio.no/sites/default/files/thumbnails/image/colourbox9783171.jpg

Bildet som er tilknyttet innholdet. Brukes som miniatyrbilde i sosiale medier og andre tjenester. Feltet er forhåndsgyldig med URL-adressen til toppbilde.

**URL-adressen til innholdet**

https://titan-app-test01.uio.no/node/1630

URL-adressen til innholdet du vil dele. Håndterer duplisert innhold.

**Rettigheter**

Alle rettigheter forbeholdt

Informasjon om åndsverk, som opphavsrett eller varemerke. Denne meta-taggen beskytter ikke nettstedets innhold eller åndsverk.

Figure 16 Basic Meta tags for Search Engines
Open Graph-taggene lar deg spesifisere metadata for å optimalisere hvordan innholdet som deles ser ut i Facebook, og i andre tjenester som Pinterest og LinkedIn. I tillegg gir Open Graph-taggene tilgang til utvidet data om brukerne som tar del i innholdet ditt via Facebook Insights.

**Innholdstype**

Artikkels og Blogg

Dette er for å beskrive hvilken type innhold det er som deles. Vanligvis vil det dreie seg om artikkler eller blogg.

**Sidetittel**

Lingvistikk, statistikk og franske hatter

Tittelen til innholdet. Feltet er forhåndsutfylt, men kan endres hvis ønskelig. Feltet er begrenset til 95 tegn.

**Beskrivelse**

Du har kansje hørt at datamaskiner kan gjøre beregninger basert på 0 og 1. Med språkteknologi, kan datamaskiner bruke 0 og 1 til å forstå menneskets språk.

Dette er beskrivelsen Facebook vil vise i skjermdumpen av innholdet. Feltet er forhåndsutfylt, men kan endres hvis ønskelig. Skriv maks 297 tegn.

**URL-adressen til forfatterens Facebook-profil**

URL-adressen til forfatterens Facebook-profil (finnes i adresselinjen på profil siden).

**URL-adressen til utgiverens Facebook-profil**

https://www.facebook.com/Titanuono-836558236423265

URL-adressen til utgiverens Facebook-profil. Feltet er forhåndsutfylt med Titan sin Facebook-profil.

**Kategori**

Teknologi

Kategorien innholdet hører til. Dette kan være Teknologi, Innovasjon osv.

*Figure 17 Open Graph (Facebook) Meta tags*
<table>
<thead>
<tr>
<th>Tagge</th>
<th>MNKOM, språktøkologi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevante nøkkeltal for innholdet, lignende til Tags-feilet. En kommaseparert liste kan brukes.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>URL til relatert innhold</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL-adressen til innhold som er relevant for artikkelen, blogginnlegget eller smånytt.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>URL til nettsiden</th>
<th><a href="https://ltian-app-test01.uio.no/node/1630">https://ltian-app-test01.uio.no/node/1630</a></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>URL-adressen til bildet</th>
<th><a href="https://ltian-app-test01.uio.no/sites/default/files/thumbnails/image/colourbox9783171.jpg">https://ltian-app-test01.uio.no/sites/default/files/thumbnails/image/colourbox9783171.jpg</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>URL-adressen til bildet Facebook vil vise i skjermdumpen av innholdet når det deles. Feilet er forhåndsutfykt med URL-adressen til toppbildet, men kan endres hvis ønskelig.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bildetype</th>
<th>JPG, JPEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bildeformat. Feilet er forhåndsutfykt.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bildets bredde.</th>
<th>5394</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bildets bredde oppgitt i piksler. Feilet er forhåndsutfykt.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bildets høyde</th>
<th>3744</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bildets høyde oppgitt i piksler. Feilet er forhåndsutfykt.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Språkform</th>
<th>Bokmål</th>
</tr>
</thead>
<tbody>
<tr>
<td>The language of your content. You may choose between English, and Norwegian Bokmål/Nynorsk.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dato og tid innholdet ble publisert</th>
<th>2015-06-09T10:49:08+02:00</th>
</tr>
</thead>
</table>

**Figure 18 Open Graph (Facebook) Meta tags II**
Figure 19 Facebook (Open Graph) Page type
**Twitter Cards**

Twitter Cards er meta-taggene bestemmer hvordan innholdet ser ut når det blir delt på Twitter.

**Type av Twitter Card**

- **Sammenkort**
- **Summary**
- **Summary with Large Image**
- **Player** (brukes for innhold med videoklipp)

**Sidetittel**

*Lingvistikk, statistikk og franske hatter*

Tittelen på innholdet som du ønsker å dele. Det er greit å bruke en annen tittel enn den forhåndsutfyltet verdien. Feltet er begrenset til 70 tegn.

**Beskrivelse**

Du har kansje hent at datamaskiner kun kan gjøre beregninger basert på 0 og 1. Med språkteknologi, kan datamaskiner bruke 0 og 1 til å forstå menneskespråk.

Dette er beskrivelsen av innholdet som du ønsker å dele. Feltet er begrenset til 200 tegn. Feltet er forhåndsutfylt som for søkemotorene og Facebook, men kan endres hvis ønskelig.

**Nettstedets Twitter-konto**

*@unioslo_titan*

Titans @brukernavn som vises i kortets bunntekst. Feltet må inneholde @-symbolet.

**Forfatterens Twitter-konto**

Forfatterens @brukernavn. Feltet må inneholde @-symbolet. Brukes med Summary with Large Image-kortet.

**URL til nettsiden**

https://titan-app-test01.uio.no/node/1630

URL-adressen til innholdet. Feltet er forhåndsutfylt, men kan endres hvis ønskelig.

**URL-adressen til bildet**

https://titan-app-test01.uio.no/sites/default/files/thumbnails/image/colourbox9783171.jpg

Bildet som vises når innholdet blir delt. Bør være et kvadratisk bilde som ikke er mindre enn 60x60 pikser. Feltet er forhåndsutfylt med URL-adressen til toppbiletet.

---

**Figure 20 Twitter Cards (Twitter) Meta tags**
Bilde som vises når innholdet blir delt. Bildet som vises skal være et kvadratiskt bilde som ikke er mindre enn 60×60 pikser. Feltet er forhåndsuttalt med URL-adressen til toppbilledet.

**Bildets bredde**

4362

Bildets bredde oppgitt i pikser.

**Bildets høyde**

2904

Bildets høyde oppgitt i pikser.

**Alternativ bildetekst**

Professor emeritus Bjørn Pedersen med den nye utgaven av Jørgen Glaesens forelesningsnotater i organisk kjemi fra 1830. Foto: Bjørn Rønja

Alternativ bildeteksten formidler formålet med bildet. Feltet er begrenset til 420 tegn.

---

**Figure 21 Twitter Cards (Twitter) Meta tags II**

**Twitter Cards**' meta-taggene bestemmer hvordan innholdet ser ut når det blir delt på Twitter.

**Type av Twitter Card**

- Ingen -
- Sammendrag
- Sammendrag med stort bilde. (Summary with large image).
- Medieavspilling

---

**Figure 22 Twitter Cards: Card Types**
INTERNASJONALE STANDERDER

Dublin Core meta-taggene følger internasjonal standard og gjør identifisering, klassifisering og interoperabilitet enklere. Dublin Core meta-taggene defineres av Dublin Core Metadata Initiative (DCMI) og benyttes ikke til direkte kommersielle formål som Open Graph, Twitter Cards, og Google Plus.

Sidetittel
Lingvistik, statistikk og franske hatter

Navnet til innholdet

Beskrivelse
Du har kansje hørt at datamaskiner kun kan gjøre beregninger basert på 0 og 1. Med språkteknologi, kan datamaskiner bruke 0 og 1 til å forstå menneskespråk.

En kort sammenfatning av innholdet. Feltet er forhandsutfylt med sammendraget.

Tema

Teknologi

Tematet for innholdet. Velg ett av nøkkelordene.

URL-adressen til Utgiveren
http://www.mn.uio.no/

Det matematisk-naturvitenskapelige fakultet er utgiveren av Titan.

URL-adressen til Bidragsyteren

Eksempler på bidragsytere kan være en person, organisasjon eller et selskap som har bidratt til å lage innholdet.

URL-adressen til skaperen
https://titan-app-test01.uio.no/user/256


URL-adressen til kilden

URL-adressen til originalinnholdet. URL-adressen til trenger ikke tilhøre Titan.

Figure 23 Dublin Core (International Standards) I
URL-adressen til kilden

URL-adressen til originalinnholdet. URL-adressen til trenger ikke tilhøre Titan.

URL-adressen til relevant innhold

URL-adressen til relevant innhold trenger ikke tilhøre Titan.

**Type**

| Tekst |

Type innholdet. Vanligvis vil det dreie seg seg om tekst, bilde, eller video.

**Filformat**

| text/html |

Innholdets filformat. Feltet er forhåndsutfylt med text/html, men kan endres hvis et annet filformat er brukt.

**Språk**

| Bokmål |

Språket innholdet er skrevet på. Du kan velge mellom Bokmål, Nynorsk, eller Engelsk.

**URL-adressen til innholdet**

https://titan-app.test01.uio.no/node/1630

URL-adressen til innholdet. Feltet er forhåndsutfylt, men kan endres hvis ønskelig.

**Rettigheter**

| Alle rettigheter forbeholdt |

Informasjon om rettighetene over innholdet. Alle rettigheter er forbeholdt som standard, men verdien kan endres hvis ønskelig.

**Dato**

| 2016-07-21T09:53:08+02:00 |


Figure 24 Dublin Core (International Standards) II
Disse meta-taggene kontrollerer hvordan innholdet ser ut når det deles på Google+, eller forhåndsvises med Rich Snippets og Knowledge Graph Cards. Rich Snippets er den utfyllende teksten som vises i søkresultatene. Knowledge Graph Cards presenterer et kort med struktureret og sammenfattet informasjon om emnet i høyre sidan av søkresultatene.

**Innholdstype**

Artikel

Typen av innholdet. Type defineres i Schema.org®, et samarbeidsprosjekt som gir utgivere og produsenter muligheten til å få ytterligere informasjon om sitt innhold synlig i søkmotoren. Fellet er forhåndsnutfylt med relevant type, men kan endres hvis ønskelig.

**Sidetittel**

Lingvistik, statistikk og franske hatter

**Beskrivelse**

Du har kanske hort at datamaskiner kan gjøre beregninger basert på 0 og 1. Med språkteknologi, kan datamaskiner bruke 0 og 1 til å forstå menneskespråk.

Et sammendrag av innholdet på maks 200 ord. Feltet er forhåndsnutfylt med sammendraget, men kan endres hvis ønskelig.

**Overskrift**


**Forfatter**

Henrik Hillestad Løvold

Forfatteren av innholdet. Fellet er forhåndsnutfylt med brukernavnet ditt, men du kan endre verden hvis ønskelig.

**Utgiver**

Denne meta-taggen er for bedrifter som ønsker å ta eierskap over innholdet sitt. Du bør lenke til utgiverens Google+-profil.

**URL-adressen til bildet**

https://bitan-app-test01.uio.no/sites/default/files/thumbnails/imagga/colourbox9783171.jpg

URL-adressen til at uriett bilde som representerer innholdet. Ikke bruk et generelt bilde, for eksempel

---

**Figure 25 Google Plus (Google) I**
**URL-adressen til bildet**

https://titan-app-test01.uio.no/sites/default/files/thumbnails/image/colourbox9783171.jpg


**Logo**

https://titan.uio.no/sites/all/themes/titan/img/titanlogo-black.svg

Titan sin logo. Fellet er forhåndsutført, men URL-adressen til et annet logo-bilde kan oppgis.

**Publikasjonsdato**

2016-06-08T10:49:08+02:00


**Modifikasjonsdato**

2016-07-21T09:53:08+02:00


**Figure 26 Google Plus (Google) II**

▼ **GOOGLE**

Disse meta-taggene kontrollerer hvordan innholdet ser ut når det delas på Google+, eller forhåndstilsvises med Rich Snippets og Knowledge Graph Cards. Rich Snippets er den utfyllende teksten som vises i søkeresultatene. Knowledge Graph Cards presenterer et kort med struktureret og sammenfattet informasjon om emnet i høyre siden av søkeresultatene.

**Innholdstype**

- **Artikkkel**
- **Artikkal**
- **Blogg**
- **Nyhetsartikkal**
- **Skapende Arbeid**

**Sidetittel**

Tittelen

**Figure 27 Google Plus (Google): Item types**

147
Google News® er en datagenerert nyhetsjeneste som samlar overskrifter fra nyhetskilder over hele verden, setter relaterte saker sammen i grupper, og viser dem i samsvar med hver leserens interesser.

**Google News Nøkkelord**

En kompakt liste av nøkkelord som hjelper Google News® å bestemme hvordan innholdet ditt skal klassifiseres. Nøkkelord kan også skilte mellom beslektede begreper. For eksempel kan man bruke nøkkelordene “VM i Brasil 2014, Spania mot Nederland, fotball” for å spesifisere at artikkelen omhandler fotball-VM i motsetning til rugby-VM.

**URL-adressen til unik kvalitetsjournalistikk**

Fremhev unik kvalitetsjournalistikk i publikasjonen din, eller gi oppmerksomhet til andre som fortjener det med Google Standout®. Unik kvalitetsjournalistikk innebærer at organisasjonen har investert betydelige ressurser i å produsere innholdet. Ikke bruk denne taggen mer enn sju ganger per kalendervekst.