Delft University of Technology

Master Thesis

Learning Object Metadata Workflows for Description, Findability and Reusability Improvement

Author:
Aneliya Dimitrova
a.dimitrova@student.tudelft.nl

Supervisor:
Dr. Christoph Lofi
c.lofi@tudelft.nl

Web Information Systems Research Group
Department of Computer Science
EEMCS Faculty, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

Learning Object Metadata Workflows for Description, Findability and Reusability Improvement

MSc Thesis

Submitted in partial fulfillment of
the requirements for the degree of
MASTER OF SCIENCE
in
COMPUTER SCIENCE
Software Technology

by

Aneliya Dimitrova Dimitrova

Student number: 4501667

Web Information Systems
Department of Computer Science, EEMCS,
Delft University of Technology,
Delft, the Netherlands

To be defended publicly on Thursday 30/08/2018 at 13:00h

Supervisor:	Dr. Christoph Lofi	WIS, EEMCS, TU Delft
Committee:	prof. Dr. ir. Geert-Jan Houben	WIS, EEMCS, TU Delft
	Dr. Christoph Lofi	WIS, EEMCS, TU Delft
	Dr. ir. Georgios Gousios	SERG, EEMCS, TU Delft

This thesis is confidential and cannot be made public until August 30, 2018.
An electronic version of the current thesis is available at http://repository.tudelft.nl/.

Contents

Abstract

With the increase of online education, a good description of learning resources has become vital for educational resource sharing and reuse. Resource description has been under the spotlight in recent years. Educational platforms can benefit from good resource organisation and description, thereby providing a higher quality of services and attracting more learners to use their systems. Furthermore, well-described resources with metadata, promote content sharing and re-use.

This work starts with an extensive literature research on metadata generation techniques and breaks the findings down to metadata types. A detailed taxonomy of metadata types, based on this research, is provided. The taxonomy takes into account properties common to these types. Second, this work analyzes the state-of-the-art metadata collection techniques in literature and real-world educational content repositories including a showcase with the TUDelft library, in order to estimate the gap of metadata employment in the field of education. Following the results of this research and based on the observation that similar steps are often performed together, a set of easy-to-follow and generic enough design patters for generating metadata was identified. These design patterns aim at assisting content authors or data professionals with filling in metadata and thereafter, allowing for feature development or improvement in the respective platforms. The patterns for metadata extraction are based on the identified taxonomy of metadata. Finally, semantic metadata is extracted as proof of concept for two of the proposed patterns. A satisfactory to a high-quality result was achieved, showing that the patterns are intuitive and the data extracted with them, can be potentially used to describe the respective Educational Resource (ER) by adding the extracted information to its metadata.

1 INTRODUCTION

In the last decade, we have witnessed an increased use of massive open online course (MOOC) platforms for online education and a substantial amount of digital resources being produced for them. These platforms are gaining popularity as many of the most renowned universities in the world provide courses on them, ensuring high-quality education and allowing for easy online access to resources anytime anywhere in the world. The presence of numerous educational resources (ER) comes forth with a challenge for discovery, sharing and re-use of the ERs and makes solving this challenge a goal for many researchers and practitioners. Having proper ER description is accompanied by a potential for enhanced data quality, time-saving and a reduction of repeatedly created identical content.

In this work, both the "ER" and "LR" terms will be used interchangeably, standing for "educational resource" and "learning resource", respectively.

ERs are created by highly skilled professionals such as academics and business experts. Metadata plays an important role in describing ERs and makes them machine-understandable and especially findable. Without meaningful metadata, the platforms storing resources, are simply repositories with resources that people out of the organization can hardly find and use.

Much attention has been drawn to establishing metadata standards, however, the practice has shown that learning resources often lack even the fundamentals of valuable metadata [1].

This work investigates possibilities to assist practitioners in the metadata generation of their learning resources and subsequently, allow platforms to improve services and resource quality, findability and re-use. To get an insight of the best approaches for metadata collection and generation, numerous scientific papers from the last fifteen years from important conferences were evaluated, such as LA-CCI[1], ICALT[2], WWW[3], ISWC[4]. Practitioners are also part of the current research, in particular several MOOC platforms were investigated and poor metadata coverage was confirmed. This was further reaffirmed by discussions with TU Delft university's library. In general, two metadata standards are suggested by most researchers and adopted in most platforms. However those standards are not adopted completely as the ER metadata is adjusted in an ad-hoc manner, i.e. based on their needs. All findings related to this matter, are further discussed in detail in sec.(3).

[1] http://la-cci.org/

[2] http://www.guide2research.com/conference/icalt-2018

[3] https://www.w3.org/Conferences/Overview-WWW.html

[4] http://swsa.semanticweb.org/content/international-semantic-web-conference-iswc

As a result of the extensive literature and MOOC platform analysis of the current state-of-the-art techniques for educational metadata (sec.3) and based on common properties of metadata elements, metadata is first split into types. Next, a metadata type taxonomy is proposed, see sec.(3.1.1). Following, generic patterns for metadata generation and collection were abstracted. Design patterns, as known in software engineering, aim to provide solutions to commonly seen problems in such a way that they can also be modified in case of future requirement changes. Following this and other guidelines [2][3] in designing patterns for solving particular problems, eight patterns have been proposed for generating metadata, based on evidence in literature and practice. The proposed patterns are abstract enough to ensure wide application and at the same time are able to extract a certain type of metadata in a way that eases the ER authors and data professionals. Having the taxonomy in mind, one can easily figure out what data they need and follow one of the proposed design patterns. The patterns aim to encourage ER authors and data professionals to fill in the necessary data by following a readily accessible set of steps.

Following discussions with the library of Delft University of technology, their requirements were evaluated and semantic metadata was extracted applying two of the proposed patterns.

The experiments intent to show that by following the decision tree that is provided in section (4), the user gets a realistic expectation of the process for extracting certain types of metadata, including the effort involved and the predicted quality of the results.

Summarising, this thesis work contributes to solving some issues like inconsistency and incompleteness [4] of metadata within the educational domain as follows:

1. Extensive analysis in literature and among practitioners regarding **State-Of-the-Art (SoA)** techniques for educational metadata generation.
2. **Identifying gaps** between research and practice in terms of metadata completeness and implemented metadata standards.
3. Support practitioners to extract or collect metadata by helping them understand the type of data they need, via **a metadata type taxonomy**.
4. Support practitioners to extract the necessary metadata type by proposing **design patterns** for every type of metadata discussed in this work.
5. **Demonstrate the potential benefit** of the patterns by applying two of them to extract a specific metadata type according to the needs of the TU Delft library which contributed with real user input.

The rest of the thesis is organized as follows: Section (2) is about related work from the recent years, dedicated to metadata generation. Section (3) discusses state-of-the-art techniques to obtain metadata by comparing literature with industry practises and discussing the gaps between them by splitting data into types and proposing a metadata type taxonomy. Section (4) is the core of the paper, together with the previous section where the proposed design patterns are first discussed in detail based on the metadata taxonomy. Their efficiency is illustrated by conducting experiments in section (5). Section (5) also describes an example practical setup, aimed towards illustrating proof of concept for semantic metadata extraction via the proposed patterns. Section (6) discusses the results from section (5). Finally, in section (7), conclusions derived from the experiments, and finally, limitations and future work are overviewed in relation to the topic of this thesis work.

2 RELATED WORK

2.1 Educational metadata standards

The growing number of educational content poses a challenge in finding the resources with highest quality. A key component to enable findability and extra functionality related to the ERs, is their metadata. Metadata is necessary for machines to process human-understandable content. It has many applications and functions, for example, when describing resource relationships, or providing more semantic information about the content of the resource. Lack of unified schema design of educational metadata causes issues such as difficult access to resources, content management issues and sharing issues. Generally, there are several approved schema standards that are used to build upon, but not a single one that is enforced to all ERs. The Learning Object Metadata (LOM) [5], a joint work between the Learning Technology Standardization Committee (IEEE) and IMS [6] (full schema in appendix {A}), Dublin Core (DC)[5] metadata, are the base standards for educational metadata. In addition to them, several application profiles (AP), such as ARIADNE [7] and SCORM[6] are commonly discussed. An application profile is a modification of a certain standard by adding and/or removing elements to/from it. Dublin Core is a general purpose and widespread standard for multiple disciplines such as fine arts [8]. DC ensures basic metadata presence. A number of authors have recognized DC as a standard also in the educational domain [9] [10] [11]. None of the elements (also called fields) of DC are mandatory, however, upon adoption, the administrator can set certain elements as optional, conditional or mandatory. Elements being optional allows for freedom when using the standard, however, on the other hand, it limits the possibility of consistent and complete metadata collection which is the purpose of using a metadata standard to index educational resources. Similarly, all LOM data elements are also optional. The difference between the two schemas is that DC contains a set of 15 elements spread into 3 classes: "Content", "Intellectual property" and "Instance" and has no hierarchy. To the contrary, LOM has 96 elements in 9 classes and is organized into a hierarchical structure. A detailed discussion of a survey from 2012 providing the whole picture of LOM standard utilization, is further discussed in section (3.3). Further on, elements of LOM and DC standards as well as some extra elements are considered when performing analysis on real MOOC platforms.

2.2 Metadata generation

Find-ability and re-use are the ultimate goals for this thesis work. To achieve them, it is necessary that each and every resource in an educational system is accompanied by high quality metadata. Obtaining this data, however, is not a straightforward process and depends to a large extent on human contribution.

Most often, the burden of metadata completion falls on the author or publisher of the resource. The manual approach, while it can help with scalability, is infeasible. While each author may have no more than several resources to describe with metadata, the metadata elements per resource can easily reach hundreds. This explains why it may be infeasible for the author to spend time on it.

Assuming that the element set per resource is not very big and the author can handle it, the simplest and most adopted way to fill in data for an element is by selecting a value from a list of pre-defined options, known as controlled vocabulary. This approach, however, limits the author and sometimes doesn't fit their needs. Thereupon, free text is an option too, yet, one that introduces a lot of noise and mistakes such as misspelled words.

Therefore, in the early 2000's researchers attempted to automate metadata generation in order to ease the work of the authors and practitioners. Automation is helpful because it allows for more data to be processed at once, mitigates the risk of human error and the produced output can be used by practitioners to implement services or improved the existing ones. Motelet et al.[12] propose automatic generation of some LOM elements such as "Semantic density", on the basis of graph building. The authors also suggest that the simultaneous editing of the same element in multiple ERs makes more sense than editing each resource separately. Motelet et al. believe that it is best to suggest an automatically extracted value for a given element, but keep the human involvement in the whole process.

On the other hand, in the early 2000s, Hatala et al.[13] also propose a final value to the author, but to extract this value, they look at their data from three perspectives: (1) individual record as is; (2) an assembly of records with a hierarchical structure which a particular record belongs to, and (3) a repository of metadata for the record. They prove their assumption that in a hierarchical structure some elements can inherit part of the metadata from their parent resource or based on the context of the used surrounding resources. Therefore by combining inheritance, aggregation, content based similarity and connecting ontologies and special logical rules, it is possible to achieve high quality results in smart metadata generation. However, the main limitation of their work is that their approach does not reduce the amount of work for the author. To solve this,

[5] http://dublincore.org/

[6] https://scorm.com/scorm-explained/

they plan to assign weights to each value in the final list and this way, to compute the final value automatically in order to reduce the workload for the author.

Additionally, Cadinaels et al.[9] draw our attention to the extraction of metadata from several sources: (1) user profiling of content creators (for elements such as author, contact information, affiliation etc.); (2) ER management systems with resources stored together having inheriting element values and lastly, Learning Management Systems (LMS) that can obtain data that provides contextual information on the usage of the ERs in different topics and courses. For instance, having several ERs covering a topic from "data science", increases the chance a new "Data science" resource may inherit part of the metadata of the existing ones.

The extraction of semantic information is often treated as a classification [14] [15], or a clustering [16] [17] problem and it is approached via machine learning (ML) models that teach a system to find and recognize the correct information for a certain metadata element. Furthermore, in semantic information exaction, ontologies are utilized rather frequently [15], [18], [19] for extraction of elements such as content's topic, prerequisites etc. Researchers also occasionally suggest crowdsourcing [20] to help with ontology creation, result evaluation, or simply to generate their own metadata by adding their own keywords or rate the resource in terms of quality and difficulty.

On the other hand, elements that provide technical information such as format or size, are easy to extract directly from the ER, and a lot of tools exist that can automate this process [21]. Tools like Data Fountains[7], Dspace[8], Omeka[9], or Editor-Convertor Dublin Core Metadata[10] are all tools that generate data either by scanning the meta-tags of HTML page (if the resource is online) or by harvesting metadata from the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). Some of them like RepoMMan[11] can also extract the title and keywords and provide the results to the user for confirmation.

The aforementioned techniques are often seen performed in similar forms by researchers. Section (4) presents design patterns that rest on commonly observed metadata extraction techniques suggested in literature and presented in this section.

3 STATE-OF-THE-ART

This section aims to investigate and present the gaps between the current state-of-the-art extraction techniques in literature and among practitioners in today's MOOC platforms with regards to specific metadata elements. The section covers metadata elements, suggested by scientists and the functionality they would serve, if implemented in MOOC platforms. This is necessary as based on the desired functionality, each individual metadata element can have a higher or lower value. For example, for ERs aimed at being used in the United States, the "language" element may not be important, but in Belgium, for instance, the "language" element has a high value due to the multi-lingual population of the country. As a result from the analysis, a taxonomy of metadata types is proposed in order to mitigate the metadata extraction process by letting practitioners understand and identify the different types of metadata they need.

3.1 State-of-the-Art in Literature

The LOM standard, being a standardized metadata schema, has received attention in a significant number of research papers for several reasons. First, it has a big number of elements describing different aspects of the Ls such as *Educational*, *Rights*, *Classification*, *Relation*. Another reason for LOM's popularity is because it allows binding via XML and RDF which are the main metadata languages used to describe data. The key problem is that while it has a considerable amount of elements they are not enough to describe the needs of every system. LOM is often insufficient to provide enough information for purposes such as a personalized recommendation [17] or modelling of learning and teaching styles [15] as these tasks require knowledge in aspects such as user knowledge and performance. These are not covered by LOM. That being said, despite the common agreement to use LOM as a foundation schema in education, on numerous occasions practitioners need to build up on top of

[7] http://datafountains.ucr.edu/

[8] http://www.dspace.org/

[9] http://omeka.org/

[10] http://www.library.kr.ua/dc/dceditunie.html

[11] http://www.hull.ac.uk/esig/repomman/index.html

the standard and/or eliminate the metadata elements that are not required for their system. [15], [17], [22], [23]

The metadata elements for ER description are selected based on the objectives of the learning management system. Capuano et al.[22] as well as the recent work of Miranda et al.[23] classify the following metadata elements[12] as important, led by their goal to personalize and contextualize learning activities:

- Language
- Domain/Concept
- File Type
- Dimension
- Learning Resource Type
- Duration
- Interactivity type*
- Interactivity level*
- Difficulty*
- Semantic Density*
- Time to learn*

The values of these elements are obtained by automatic analysis of the technical details of the content as well as by trying to combine learning models, statistical analysis and ad-hoc heuristic rules to extract element data such as ”interactivity type” and ”interactivity level” from ”MIME type” element. For the educational elements, Miranda et al. base their heuristic rules on pedagogical teaching and learning principles described by Bloom [24] and Ronsivalle [25].

Particular researchers attempt to make use of the meaning of the data and try to describe the resources semantically. Farhat et al. [15] use the LOM standard as a base. By having an input of LOM metadata as per three criteria, they output semantic metadata that describes the content of the resources. The criteria are: (1) the LOM elements they consider, must relate to the educational content of the ER; (2) the elements must be required in most of the LOM application profiles and (3) the element's data must be filled in in most of the application profiles. By defining the above-mentioned requirements and incorporating ontologies they created, they ensure extraction of the most valuable information for the ERs. In the end they claim high accuracy of extracted semantic information for *”title”*, *”description”* and *”keywords”*.

Recent research from 2017 by Othman et al. [26] shows that knowledge extraction is able to improve search results by extracting metadata from existing videos. Othman et al. do not build their own metadata schema per se, however they do extract technical, ”web” and descriptive metadata automatically from the videos. Since the technical metadata is easy to extract automatically, the focus is mostly on the descriptive data of scenes, shots, objects etc. The web metadata includes some elements from the LOM schema under the Technical and the Descriptive categories, therefore they can be unified into technical and descriptive according to their function. Table (I) shows the original elements categories as per the author's categorization.

Technical	**Web**	**Descriptive**
File name	URL	Scenes
Duration	Views	Shots
Resolution	Likes	Objects
Bitrate	Dislikes	Places
Frame count	Comments	Summary
Frame width	Tags	
Aspect Ratio	Ratings	
Quality		

TABLE I. *Metadata elements in [26]*

This element choice of the authors and the successful extraction of the information, is another example of technical metadata extraction by using metadata extraction tools, followed by classification and clustering machine learning techniques to classify videos into topic categories and to extract also other parts of the descriptive metadata.

Focusing on the learning style of the MOOC participants, it is possible to create personalized content based on several educational LOM elements and based on the sixteen learning styles defined by Coffield et al.[27]. In a recent study (2017) Dorça et al.[17] investigated and proved that it is possible to achieve personalization of ERs by clustering, combining learning styles and the LOM elements, listed below in order to suggest only content, suitable to the student.

- Structure
- Format
- Interactivity type*
- Learning Resource type*
- Interactivity level*

[12] LOM *Educational* elements are marked with *

Following this element choice, it

Despite the wide usage of LOM, it is not as widely spread, as the Dublin Core (DC) standard. Due to its small size with very general elements, it is much easier to implement in a system that does not intend to have much functionality and information about *title*, *resource type* (video, audio, document etc.), *author*, *subject*, *keywords* would suffice. Again, this is based on the objectives of the system.

Catarino et al.[10] and Halpin et al. [28] focus their work on folksonomies[13] and identify new interesting and important metadata elements that are not part of the Dublin Core, but could be a valuable addition to it. By attempting to match existing DC elements with folksonomies, they explore new elements, that folksonomy tags did not have a match for in the existing element set, namely:

- Action (toRead; toPrint etc.)
- Category
- Didactic intent (overview, explanation etc.)
- Rate (veryGood, Excellent, Poor etc.)
- User name
- Utility (custom tag, e.g. teacher name, concept etc)
- Notes
- To be used in (work, university, school etc.)

Elements not having an existing match within the element set of DC forced the authors to create these extra elements in order to describe the data. This approach has an advantage of being "crowdsourced", that is, they utilize real user input into the metadata elements and prove that DC is not sufficient to describe all aspects of the ER.

Going further with the used standards, the ARIADNE schema is also seen in literature, however not as common as LOM and DC. The ARIADNE metadata schema is a predecessor of the LOM standard. In a cooperation between the K.U Leuven in Belgium and the ARIADNE project members, the so called Knowledge Pool Management system emerged which offers interesting means of handling metadata in terms of elements choice [7]. Here, very interesting semantic and educational data is included on top of the general descriptive and technical metadata elements. The semantic and educational elements are listed below:

- End user type
- Document type
- Didactic context (target learners)
- Course level
- Difficulty level
- Semantic density
- Pedagogical duration
- Discipline
- Main concept
- Main concept synonyms
- Other concepts

The elements *"Didactic context", "Discipline"* are said to be restricted with a controlled vocabulary in order to escape ambiguity especially in the discipline case, where authors suggest that synonyms for the main concept can be provided in order to support more languages. Some of the above listed elements are not part of LOM, even though ARIADNE is its predecessor, showing how the elements are selected on basis of their usefulness and not by adopting a single standard. This fact explains the high number of application profiles seen in research and practice.

This aspect of the research suggests that a lot of effort is put into extracting semantic metadata and that this type of metadata can indeed improve educational platforms from many perspectives. Generally, technical elements and some of the descriptive elements are fairly easy to extract, however that is not the case with pedagogical and educational data which aims to provide personalization of content, similar content suggestions, to adapt the content to the learning style of the user etc. This is still a challenging task.

3.1.1 Towards metadata type taxonomy

Due to the wide variety of applications of metadata for educational resources, the knowledge found in literature is organized and a metadata type taxonomy was formed as a result from that. This taxonomy divides metadata into groups based on their type and common features. The main two features for defining the proposed taxonomy, are the **purpose** of the metadata (e.g. for education, administration, description), and based on the **extraction method** of the metadata element (e.g. objective, subjective, intrinsic, semantic). Metadata elements whose values can be directly extracted from the resource entity, are called "intrinsic", while "semantic" elements require analysis of the content of the educational

[13] https://en.wikipedia.org/wiki/Folksonomy

resource. The taxonomy is split into these two major feature groups based on observations in metadata elements selection for specific purpose and based on how researchers attempt to collect or generate the values of these elements. The taxonomy aims to present a clear picture of how these elements can be split as per their functionality or according to the extraction method and the values they can hold. (e.g. single-option values or arguable values).

A unifying diagram of the types of elements with examples, based on the aforementioned features, can be found in fig.(1).

To shed light on the diagram, "technical" metadata is a functional type while "intrinsic" metadata is a type based on the method of extraction. To a large extent the elements that fall into these two categories, overlap, because technical elements can be directly extracted from the resource entity (which is what "intrinsic" means). Most (if not all) of the technical metadata is objective, that is, the value cannot be argued. For instance, if we have an audio file, it could either be in .mp3 or in .wav *format*, but never in both .mp3 and .wav for the same resource. This makes the meta element "objective". Technical metadata is easy to extract directly from the resource entity itself. On the other hand, "Subjective" metadata is the opposite - the value it holds is often based on the opinion of people or other criteria. One such element is *Difficulty level*, which falls into the "educational" functional type, "subjective" type of extraction as well as "pedagogical", according to the function of integrating pedagogical information into the metadata. That is also the case for the "intrinsic" type.

The proposed taxonomy presents the first step towards the design of metadata creation and extraction patterns. The main benefit of having such a structured taxonomy of element types, is that the practitioners can easily figure out what type of data their selected metadata elements should contain and as a consequence, will be able to approach the problem of extraction in the correct away and save themselves time and effort in designing all this from scratch.

3.2 State-of-the-Art in MOOC Platforms

An extensive analysis on metadata collected was performed on most of the currently trending MOOC platforms: starting with TU Delft Library, next were Coursera, Lynda, edX, Udemy, Udacity and Khan Academy. State-of-the-art in MOOC platforms was analysed in Oct 2017 via their own APIs where possible and via manual analysis of their online content and the published educational resources. To ascertain whether or not the MOOC platforms follow a certain metadata standard, a personal request was sent to their emails asking to be referred to members of their development teams who could provide more information on this matter. Unfortunately response was only received from two out of seven platforms, and none of them were able to connect us to the party that would be able to shed light on the matter, except for TU Delft Library. The case of the library is discussed in detail in section (3.2.1). In table (II) an overview of our approach towards surveying each of the platforms, is provided.

	Manual analysis	Analysis via APIs	Direct contact
TUDelft Library	✗	✓	✓
edX	✓	✓	✗
Udemy	✓	✗	✗
Udacity	✓	✓	✗
Lynda	✓	✓	✗
Coursera	✓	✗	✗
Khan Academy	✓	✓	✗
Future Learn	✓	✗	✗

TABLE II. *Types of Analysis on MOOC platforms*

The investigation continued by splitting the metadata according to the taxonomy presented in sec.(3.1.1) w.r.t. the functionality of the metadata, e.g. "Descriptive", "Educational", "Administrative" etc. Aiming to get an approximation of coverage, all the elements found on the platforms were collected in a table and put together. They were then compared and values that are part of the schemas for each platform were added in order to get the coverage. It was noted that across different platforms, some elements hold identical information, but the elements are named differently. For example "MIME type" and "resource type" hold the same information, as well as "affiliation" and "organization", "typical learning time" and "Expected duration", "key" and "id" and so on. These duplicates had to be unified into a single element for the final presentation in appx. {D} &

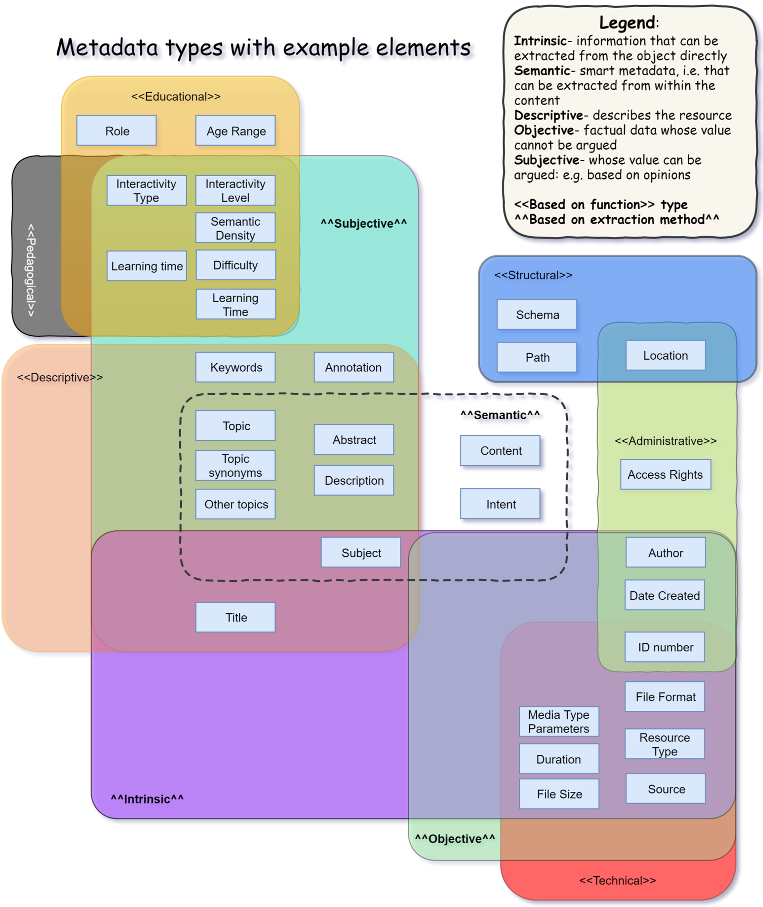


Figure 1. *Metadata type taxonomy with examples per type.*

{E}

The result of this study is compared with the suggested element set from the available literature. Discussions on our findings continue in detail in section (3.3). Please refer to appendices {D}, {E} and {F} for more a informative visualisation of our findings.

3.2.1 TU Delft Library showcase

In relation to a joint project between the Computer Science faculty and the TU Delft library, design and implementation of a federated search engine for open educational resources (OER) was discussed, including ideas and feature requirements, similar to ARIADNE's finder tool discussed in [29].

There were several desired features for the project, as follows:

1. Simultaneous search in multiple OER databases
2. Internal re-ranking and minimizing the final result into a small subset of OERs
3. Providing information on the OER such as quality, format, license, usage,
4. Possibility to segment video materials into parts defining each part with its intent
5. Content suggestions based on previous searches and usage
6. Search based on one or multiple facets (metadata elements)

Originating in the feature analysis, the necessity of certain metadata elements for the system were confirmed. Some of the elements like "quality", "language", "date", "DOI", "start time", "end time", "resource type"/"format" and "didactic intent" were missing from the current ERs of the university. These elements could be further split according to the proposed taxonomy into technical and educational from a functionality perspective, or according to the extraction method - into "semantic" and "subjective". By discussing the metadata TU Delft currently utilises, more evidence of poor metadata platform implementation was found, due to the fact that the library currently implements some (the most general) of the elements of the Dublin Core schema for the ERs they store and exploit at the moment.

Furthermore, the TU Delft library addressed an important question with respect to the project, namely how to collect existing metadata from the OER databases where the resource is stored, rather than generating it from scratch. This is called metadata schema integration or metadata cross walking. Even though that is not the core topic of this work, it attempts to shed light on it in the design patterns section (4). An additional design pattern was included (sec. 4.9) in a more generic manner - by outlining general approaches of integrating metadata schemas.

Further on, this work contributes to the TU Delft project by extracting semantic metadata for the "**Didactic intent**" in the experimental part (sec. 5), by following the proposed design patterns from section (4). Results from the experiments, are further discussed in section (6).

3.3 Gaps between literature and practice

In this section, suggested elements in literature are compared, providing results from the investigation done on real MOOC platforms - presented in sec.(3.2). The elements are matched to the taxonomy of types, proposed in sec.(3.1.1).

Overall, regardless of the exact approach towards metadata extraction, once available, the resulting values are fed into the system in one of the following ways:

1. **Free text:** by ER author at time of creation
2. **Free text:** by ER user at time of exploitation
3. **Pre-filled value:** as a result from an information extraction algorithm
4. **Categorical value:** by ER author at time of creation via provided *controlled vocabulary*
5. **Numerical value:** by ER author at time of creation
6. **Averaged numerical value:** based on collective user input

According to a survey[14] on the utilisation of metadata standards, conducted in 2004 by N.Friesen, the employment of educational metadata is not high. Overall, at most half of the LOM elements are exploited in systems, most of which also persist in the DC standard. The most precise and complete one being the usage of the "Classification" class from the LOM standard. These statistics, as of 2018, can be confirmed by analysing the current most trending MOOC platforms. It comes out that most of the commonly utilised elements persist in the Dublin Core element set. Following the process of our study in sec.(3.2), the support of metadata on different learning platforms was compared against each other and the results were aggregated in appendix {F}. For example, if we take the coverage per metadata type, only one out of 7 platforms, admittedly covers user-related information in their metadata (for learners). Generally, as visible on fig.(4), administrative, technical and descriptional types are covered most, albeit the coverage is less than or around 40%.

From the investigated platforms, the most well described ones were edX and KhanAcademy, as visible in fig.(3) and there is room for improvement for the TU Delft library, as visible in fig.(2). This could be explained with the fact that the main type of resources TUDelft holds, are research papers in PDF format and videos in a system, created especially for the students of the university, while edX is a MOOC platform with many courses and different type of resources where findability of courses and resources is vital for the universities that author and upload those resources. edX maintains a balance of metadata types, however as quantity it covers around 50% of the elements researched in this work per type. TU Delft, on the other hand has mainly administrative metadata for maintaining the ERs written by professors and students.

At the same time, while the platforms provide mostly high quality courses, there is currently not too much functionality that would justify the need for collecting hundreds of metadata values. Furthermore, semantic metadata is still challenging to extract and often requires knowledge and experience with different information extraction techniques. Due to the fact that online education is growing and provides good opportunities to the

[14] https://slideplayer.com/slide/5147365/

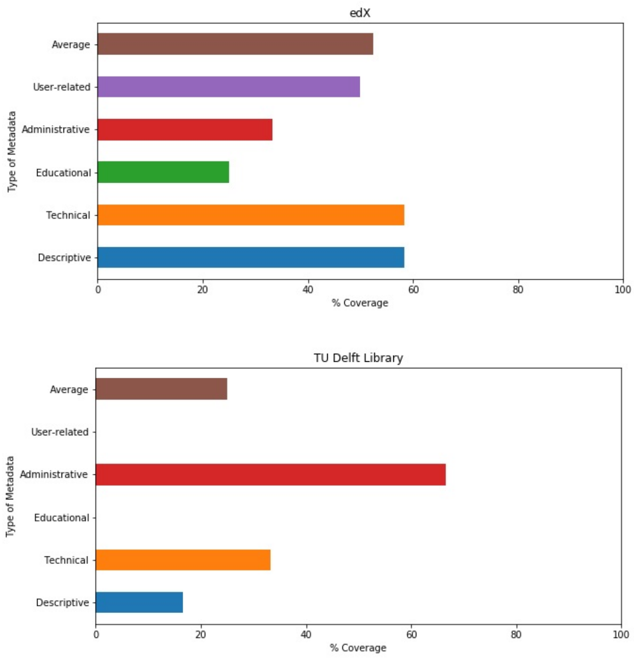


Figure 2. *EdX vs. TU Delft coverage per type*

learners - an increasing number of universities are switching to the so called ”flipped classroom”, where students work on lectures in advance and only go in class to complete exercises and to ask questions. The approach of following all lectures and assignments online and going to class after that, would increase the usage and popularity of the MOOC platforms and therefore, platform developers and owners may be better motivated to collect detailed, consistent and informative metadata in order to improve their services.

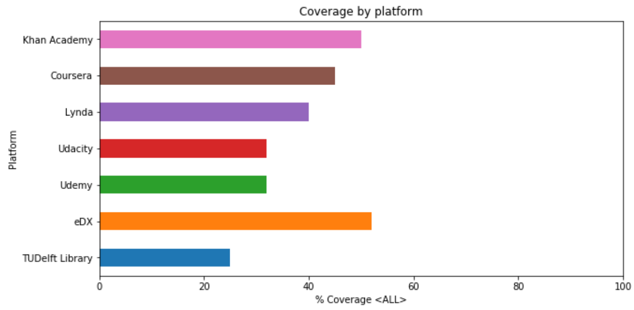


Figure 3. *Coverage by platform*

To conclude on the metadata coverage, it can be said that currently a big gap exists between research and practice and filling it may contribute not only to service and feature improvement, but also to content share-ability, re-use and interoperability.

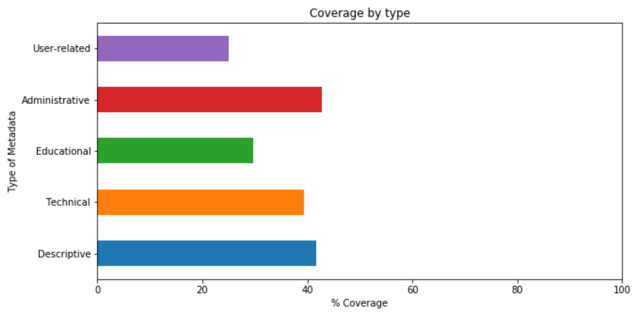


Figure 4. *Coverage by metadata type*

4 DESIGN PATTERNS FOR METADATA EXTRACTION

This section discusses in short design pattern principles in the domain of software engineering and adapts the concept of design patterns to the domain of education by aggregating commonly seen metadata extraction or collection processes into easy-to-digest steps for the practitioners to follow and get the data they need in an easy-to-follow manner. The patterns are generic design patterns that often combine both human and computational power. They cover collecting values for the most commonly utilised metadata elements as well as more complicated ones, based on their type and allow for the reader to make an informed decision.

4.1 Why Design Patterns?

Design patterns, mostly associated with software engineering, present sets of syntactic notations, a set of rules on how and when to use a pattern and advantages and disadvantages of using them. Furthermore, design patterns can be considered as a set of micro-architectures due to their contribution to the overall system architecture. Their idea is to help developers plan the design of their system based on the problem the system needs to address and solve. They depend on a particular goal and offer steps towards achieving it. Similarly, in the domain of education, the existence of such patterns would save time in the process of metadata extraction by providing a solution to practitioners that they can simply follow instead of having them spend a lot of time trying to figure that solution out.

According to the Gang of Four (GoF) [3] the goal of design patterns is to help design become more flexible, reusable, elegant and readily accessible. Patterns are abstract enough, i.e. don't provide too much detail, ensuring their relevance in a wide number of situations. At the same time they may provide a notion of problems that may occur when applied.

In general, patterns emerge from practical experience, i.e. from studying what people have already done in real systems. Martin Fowler[30] had the idea of gathering and modelling knowledge towards design patterns creation, saying that..

> "..an idea that has been useful in one practical context, will probably be useful in others."

During the literature research in sec.(2) and (3), a need for such design patterns in the education domain was identified, due to multiple attempts of researchers to extract identical metadata information with similar methods in order to describe learning resources. Papers from conferences, such as LA-CCI, ICALT, WWW, ISWC etc., were studied and commonly occurring methods of metadata generation patterns are pinpointed. *Generic design patterns* are proposed stepping on this research by first splitting the metadata elements in types which allows for more intuitive and logical pattern design.

The current work's derived patterns attempt to give steps towards decision making, rather than specific implementation steps. In the following sections of this chapter, and adhering to GoF's pattern components, example data that can be collected with each pattern, is provided together with a diagram to illustrate it as well as motivation, structural description, advantages and disadvantages of using the proposed pattern. Without using design patterns, one may miss opportunities and spend more time on trying to find a solution to a given problem, instead of having this solution immediately and only thinking about the exact way of implementing it. If the user wants to quickly find the most appropriate pattern from all proposed ones, they need to get familiar with the provided metadata type taxonomy and compare the metadata elements with it and then follow the decision tree in fig.(5) prior to checking the patterns.

4.2 Pattern: Routine Gateway

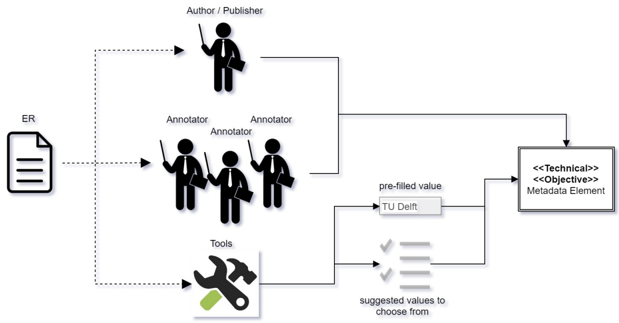


Figure 6. ***Routine Gateway*** *Pattern*

Data Types & Elements

Technical; Objective; Intrinsic

- Type
- Format
- Size
- Duration
- Views
- Path
- Location
- Number of pages
- Identifier / URL
- Creation date / time
- Start / End

Motivation, Positives and Negatives

The easiest to collect and most widely available metadata types, are technical and objective, which often overlap. That is, the data can have only one value and this value cannot be argued, such as resource size, format etc. This type of data is essential for resource management and is standardized in most metadata schemas like LOM and DC. Routine Gateway pattern on fig.(6) has the task to guide practitioners to the best fitting option by discussing advantages and disadvantages of each approach for objective metadata generation.

Most commonly this information is filled in three ways: (1) manually, either by the author or publisher; (2) manually, by a specially hired person or group of people (annotators) for this purpose or (3) algorithmically or by using tools.

The author knows best their resource content and the most intuitive way of filling this metadata would be for the author to do it. For a small number of learning resources that would be acceptable, however often the metadata elements can

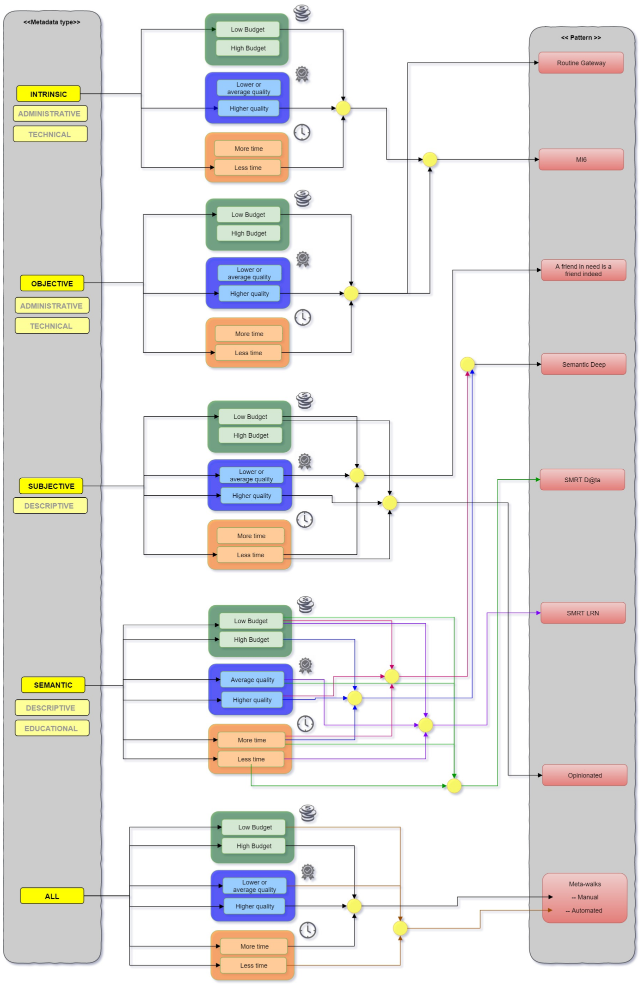


Figure 5. *Decision tree for pattern selection (Big image available in appx. {C})*

reach hundreds [31], making manual data filling infeasible. The second approach with specially hired team of annotators, while it may help with scalability as more people could do more work, is in general infeasible due to the typically large amount of resources and metadata elements in a system with ERs. Furthermore with the increase of the content, more annotators would be required, introducing more costs. Consequently, approaching this problem algorithmically gains more interest. There are multiple potential benefits in incorporating computing power via algorithms or available extraction tools to give authors a hand with metadata completion. The high performance of algorithms would make it possible for teachers to concentrate on content creation and not on metadata composition and would prevent extra costs. While a fully-automatic generation of all metadata cannot be discussed at this moment, authors can benefit from generating technical, intrinsic and objective metadata.

Requirements

Firstly, the metadata elements for which this pattern can be used, must be of technical, intrinsic or objective type. For instance, the pattern is useful for elements such as *duration*, *location* in the system, *format* etc. Following, an expert is needed to evaluate how the data will be stored, how important the necessary metadata elements are and how much metadata inconsistency can be tolerated. Often the task of manual filling of metadata seems without direct benefit for the author, therefore they may skip filling out some of it. For higher consistency, algorithmic approach is preferable. A fairly common solution to this issue is the use of metadata extraction tools such as Omeka, DSpace etc. In the case when there is need to collect metadata from files in multiple different formats, it is best to use a tool which can recognize the ER format and automatically collect the data instead of having an expert to implement algorithmic solutions for each format separately. Algorithmic solutions are the most optimal ones due to the fairly straightforward process of extraction of technical metadata. Using tools, however, can introduce other issues. One issue with tools is that they are often developed to satisfy specific needs of a certain practical or research project and cannot be applied for all purposes or metadata element extraction. That fact potentially shrinks the possibilities of utilising some of the tools. Tools like SAMGL[15], requires context knowledge about the resource, which limits the possibilities to use it. Therefore, a special requirement of this pattern, is a thorough research on the available algorithms or tools and their requirements, when algorithmic approach is considered. This task needs to be performed by skilled professionals with certain knowledge such as suitable algorithms and available tools on the market.

Structural design description

The first step in the process of the intrinsic metadata extraction, is for a decision to be made for the way metadata will be filled. When the author is going to fill it in, it should be considered that often without direct benefit, the author may skip filling some particular data or may add incorrect information. Higher budget would be needed when a group of annotators is used.

Finally, when an algorithmic approach is used, an expert needs to consider possibilities in terms of algorithms or tools and a developer is required to implement or incorporate the technique selected. Commonly, there are two types of algorithmic extraction of intrinsic metadata values. Metadata harvesting, being one of them, is widely incorporated by researchers and tools. The data is collected by harvesting the META tags [32], [33], [21], [34] within or attached to the learning resource. Once harvested, the data is fed into the respective metadata elements. The second type of automation of the collection process, is to apply some of the information extraction algorithms described in [35] on old or newly created learning resources. Once extraction is completed, the value is suggested to the user, or pre-filled.

4.3 Pattern: A friend in need is a friend indeed

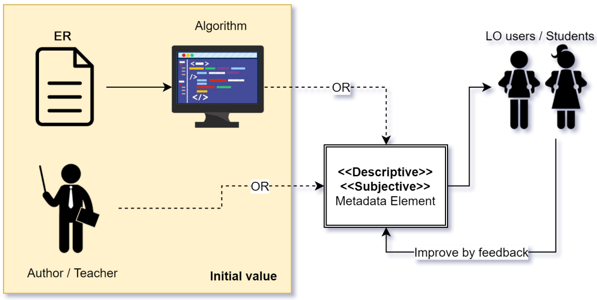


Figure 7. ***A friend in need is a friend indeed*** *Pattern*

Data Types & Elements

Descriptive; Subjective; Educational

- Difficulty
- Semantic Density
- Keywords
- Prerequisite Knowledge

Motivation

When taking an online course the student's perception differs from that of the teacher in regards to a material's difficulty level. For a teacher, a certain topic may seem easy due to the years of experience practising and teaching the topic, which introduces a bias. For a student, that material could be more difficult for various reasons such as lack of prerequisite knowledge, lack of effort and wish to learn the topic, lack of time etc.

This pattern (fig.7) aims to help improve the objectivity of metadata in LMS, because it is important for the learner to get a real unbiased idea of the quality or other aspect of the learning resource. Some metadata elements such as "Interactivity level" or "Difficulty level" of the resource are highly subjective, therefore collecting this information could be used by suggesting values to the author, that are fed by the learners, e.g. by incorporating human feedback to the metadata generation process. However the initial probable value is fed by either the author, or by an algorithm, that can extract it from related learning resources [12], [36]. Allowing user feedback improves the objectivity. User feedback on its end, could be used for example, towards adapting learning content based on the

[15] http://hmdb.cs.kuleuven.be/amg/ Download.php

learning style of the student [37].

Requirements

For this pattern to work, the initial values must be provided by the learning resource author or extracted algorithmically. In order to prove useful, the pattern should be utilised only for subjective metadata elements such as the example ones given above. Next, a developer or expert needs to be present who can implement the feedback system or module and would allow the learners to start providing feedback. Thirdly, once in a while the author will need to amend the values based on the suggestions or this could be done mathematically on the basis of majority vote or another criteria.

Structural design description

First, one needs to decide how the initial value of a specific subjective element will be supplied. If this is done algorithmically, an extraction algorithm from relevant materials can be implemented or extracted automatically based on technical parameters like size or format [38]. When the user provides feedback on the difficulty of the course or any other metadata subjective element, they improve both the learning resource and its metadata. Incorporating real user feedback is of benefit to everyone – the user, the author, the system and thereafter, the whole community of people working towards high-quality education.

4.4 Pattern: Opinionated

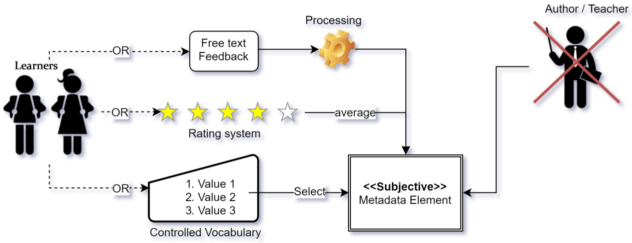


Figure 8. ***Opinionated*** *Pattern*

Data Types & Elements

Subjective

- Quality
- Interactivity Level
- Keywords
- Typical learning time
- ER-specific statistical information

Motivation

Similar to the ”A friend in need” pattern in sec.(4.3), the ”Opinionated” pattern (fig.8) incorporates feedback, however providing more possibilities and discussing their applicability for subjective metadata elements. People’s judgement of quality about a certain educational resource or an online course, is often based on the opinion of others who have already followed that course. People decide whether to download a movie or buy a product or enrol to a MOOC based on the average rating others have given it or by reading reviews. Assessing the surrounding world is a natural thing, therefore it needs to be, and often is, incorporated into the e-learning systems. Having this data to describe a learning resource, practitioners can take decisions as to whether they need to improve the quality, whether to add a new feature, what to improve in their courses etc. The current pattern provides several ways to allow users to assess the way they see a certain ER and to incorporate this into the metadata of that ER.

Requirements

This pattern must be used to subjective type of (meta)data and can be also used on some educational metadata such as *Quality*, *Interactivity level*, *keywords* etc. where data necessary can be provided in a quick way via either free text feedback, a rating system or a controlled vocabulary with specific values for the learners to choose from.

Structural design description

To gather the necessary data for subjective or educational meta elements, the users are provided with an opportunity to express their opinion within the e-learning system or in another form, for example with a survey or a game etc. The results are processed, e.g. by selecting the top N most mentioned words in the case of free text feedback, or computing the average of all-user-feedback and feeding that into the respective meta element. In the case of a controlled vocabulary with restricted set of options is used, the highest voted value is updated into the meta element. This metadata elements can and should be updated with every next vote.

Advantages & Disadvantages

A strong point of generating data this way is that it reflects the true opinion of the people regarding a certain learning resource and this data can be easily obtained and/or calculated as an average based on multiple votes or frequency count. The disadvantage is that the data changes and re-evaluated information needs to be reflected upon in the metadata of all resources that make up the learning resources.

4.5 Pattern: MI6

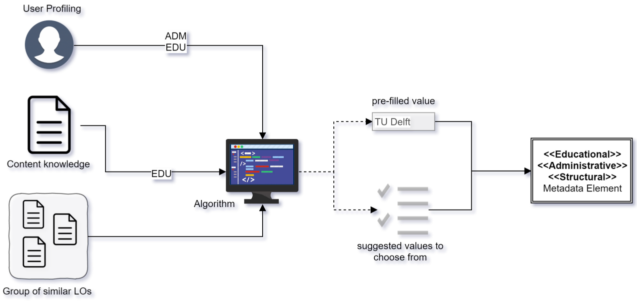


Figure 9. ***MI6*** *Pattern*

Data Types & Elements

Educational; Administrative

- Author name
- Email
- Institution
- Language
- Access rights
- Age Range
- Target Audience
- Quality

Motivation

The MI6 pattern in fig.(9) can be applied to the general elements that are required to be completed mostly by the learning resource author, because they mainly fall into the administrative or the educational type of metadata elements. Such elements are author name, institution, email address, access rights and information like field of education, category etc. Although it seems only natural for the author to complete this data manually, it can be eased by providing suggestions, or pre-filling some values with a provided option to alter them, if they don't relate to the current resource. This pattern proposes that via user profiling or via knowledge about the resource, data can be extracted and suggested to the user. An approach like user profiling or knowledge about the previous content of the same author, significantly reduces the effort of the authors, allowing them to focus on creating more content, rather than slowing them down with time-consuming and tedious metadata compilation.

Requirements

To use the pattern in its user profiling version, it is required that specific author-related information must be provided by the author. It could be collected upon system sign-up or collected with a survey, or in some other way. This data can be then used for the purpose of describing the user's resources via the metadata. To use the content knowledge and grouping versions of the pattern, algorithm(s) should be implemented to collect information such as author's previous resources, or resources, related to the current one being described etc. This allows for relevant information extraction and suggesting it to the user later. For either version of the pattern, the only requirement is to have a skilled expert at hand, who will be able to decide what will be the exact content-related extraction approach and to implement it.

Structural design description

Learning resources are often grouped by topic, author or another type of information. Based on this grouping, or assembly, specific metadata elements can be suggested for a newly created learning resource. For example, assuming that a given ER has the "Computer Science" category, a newly created ER stored together with the rest of the computer science resources, will have the same value for the "category" metadata element. Therefore, the first step is to decide which of the methods is the most suitable: content related, grouping or user profiling. They can be combined together in order to increase the range of elements that can be extracted. User profiling is useful approach for author-related metadata. Most authors have a certain affiliation, provide access rights for their material, upload resources that are in a certain language and have other types of personal information. For that reason, user profiling would be useful in helping to gather the respective administrative and/or educational metadata elements for the author. These values are either pre-filled or suggested and the author would be able to select or change them, as needed.

Advantages & Disadvantages

Having to manually fill a lot of information introduces inconsistency, errors and completeness issues. Therefore, once author information is present in the system, it can be utilised and used to help the work of the author and improve their productivity. However the usefulness of this metadata depends on the veracity of the data provided by the author. A disadvantage is that the author-related data is not present in the first place. A possibility exists that the author provided information may change with time.

4.6 Pattern: Semantic Deep

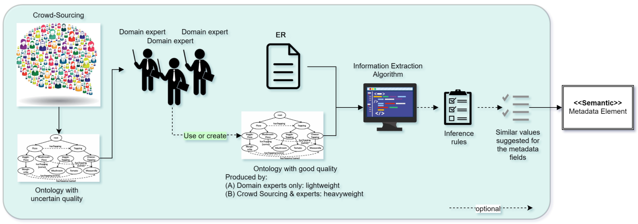

Crowd-Sourcing
Domain expert
Domain expert
Domain expert
ER
Information Extraction Algorithm
Use or create
Ontology with uncertain quality
Ontology with good quality
Produced by:
(A) Domain experts only: lightweight
(B) Crowd Sourcing & experts: heavyweight
Inference rules
Similar values suggested for the metadata fields
<<Semantic>> Metadata Element
optional

Figure 10. ***Semantic Deep*** *Pattern*

Data Types & Elements

Semantic

- Prerequisite_Knowledge
- Topic

Motivation

To allow for higher quality of the extracted semantic information, many researchers propose using ontologies [39] [19] [40] [41]. Ontologies comprise the knowledge of one or multiple domains by introducing their relationships in a computer-understandable way. Ontologies, however, are not only limited to domain knowledge, but can also express the structure of a resource management system providing aid to extract useful semantic information based on criteria like resource relatedness, location etc. Using this information, metadata can be significantly enriched. With the Semantic Deep pattern (fig.10), many potential advanced functionalities could be introduced or improved, such as prerequisite knowledge suggestion, or organizing ERs into semantically annotated LR sets and delivering them to learners on demand or based on their learning style [42].

Requirements

First and foremost, a suitable ontology is required depending on the functionality for which the metadata is being extracted. Secondly, a textual representation of the learning resource is necessary, e.g. video subtitles, lecture notes etc. When none of the available ontologies fits the needs of the user, creating an ontology needs to be considered. Ontology creation is a time-consuming task, however in the long term it is very useful. Commonly, the domain specific ontologies are lightweight and are upgraded in the process of exploitation by experts by adjusting them according to their needs. Lightweight ontology means it is less expressive and covers less knowledge in terms of e.g. concepts and their relationships as opposed to "heavyweight" ones. Next, when ontology creation is considered, a team of experts needs to be available or, alternatively, this task can be handled by crowd-sourcing. When the ontology is to be produced by experts, tools such as Protégé[16], Fluent Editor[17] and NeOn Toolkit[18] can be utilised for the process. In the end, an ontology will be available in some of the standard ontology languages like RDF. On the other hand, exploiting the notion of human-computation for the sake of solving complex problems that require more than computer power, makes crowdsourcing an excellent approach towards ontology creation. Picking the right pool of people for the task is essential as this can significantly decrease the amount of time spent on the task and increase quality. The people used for the crowd-sourcing task can be motivated by the right incentive, such as monetary [43] or non-monetary award, like common goal.

Further on, once the ontology is available, a skilled expert is required to select how to extract the data with the aid of the ontology, in other words, to select the extraction algorithm and a developer should implement it. To summarise, for this pattern the most important thing is a suitable ontology and expert to select an algorithm

[16] https://protege.stanford.edu/
[17] http://www.cognitum.eu/semantics/FluentEditor/
[18] http://neon-toolkit.org/

for extracting the metadata as well as a developer who will be able to implement this and make the most out of the ontology and the algorithm, combined together.

Structural design description

This pattern can be utilised from the beginning, with the ontology creation, when no ontology fits the needs for the specific project, or can be directly applied in combination with a selected algorithm. Due to the fact that often ontologies lack specificity for a particular domain, creating a new one is often necessary. When the essence of the necessary ontology allows it, and the task is crowd-sourced, the crowds can produce a bigger ontology, but with uncertain quality. That, on its end would need to be proof-checked by the (domain) experts to ensure or improve the quality. Finally, the output values are suggested to the user or pre-filled

Advantages & Disadvantages

Regrettably, the available ontologies are often not specific enough or do not cover some domains at all. Therefore, experts often need to take part in the creation of new ontologies for specific domains. This is costly and for a high-quality ontology, that requires specific experience or knowledge, a high budget will be required and it would cost more time and effort. On the other hand, sometimes a general purpose ontology can be utilised, which yields average quality results with less costs and effort. [13]

4.7 Pattern: SMRT D@ta

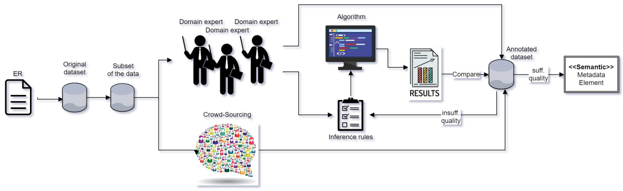


Figure 11. ***SMRT D@ta*** *Pattern*

Data Types & Elements

Semantic

- Prerequisite_Knowledge
- Topic
- Subtopic
- Concept
- Didactic intent
- Keywords

Motivation

Semantic metadata extraction can rarely avoid human involvement, because it needs to be examined at content level. This is due to the fact that learning resources often have different structure and specifics such as video subtitles that contain the speech of the narrator, together with the specific time it occurs. Another example are slides with titles and bullet points. This makes it necessary for an expert to analyse the information on a subset of the resources in order to find commonalities among the resources and to suggest extraction rules for the whole resource collection. The use of inference rules for metadata extraction is commonly seen in literature [13], [14], [23], [44]. That is, a set of a logical formulas which take premises, analyse their syntax, and return a conclusion. Therefore, this pattern relies on human-crafted rules as it attempts to make the metadata extraction process more easily applicable without the need to understand and implement complex machine learning algorithms. Some researchers use layout models for extracting information [44], however these are not universally applicable and as such, are error-prone. Therefore, this method is suitable for individual collections and can hardly be applied to learning resources of different types without initial analysis. Nonetheless, adopting the current pattern in fig.(11), utilising inference rules can promise good results on the individual ER collections when analysis is done carefully and can save time to practitioners with semantic metadata extraction.

Requirements

The main requirements for this pattern to be applicable is to have a subset of the resources, whose content needs to be analysed and to have skilled expert who will be able to do the analysis and hand-craft the rules that will be used for the extraction process. It is crucial for the right expert to take on this task in order to create accurate inference or heuristic rules. Moreover, the pattern doesn't require the resource dataset to be huge, it can be small to middle size and should be manually annotated. Depending on the required skill the manual annotation task, it can be outsourced to a pool of people or performed by experts at hand. Crowdsourcing it would not only reduce costs, but could also drastically decrease task completion time because of the large number of human power used in the process. The produced data is used to evaluate the result of the rule-based extraction. Additionally, there must be an expert available who can consider the data pre-processing steps prior to rule implementation, such as stop-words removal, punctuation removal, stemming, lemmatisation, part of speech tagging etc. This could also be the developer that would later implement the rules. To summarise, the requirements include: (1) human expert to handcraft rules based on content analysis of the ERs; (2) skilled developer to implement the rules; (3) a group of people (crowdsourcing or experts) to manually annotate the initial dataset for the evaluation stage later.

Structural design description

The structuring of the pattern starts with the expert who analyses the content and extracts the commonalities between the resource files under the form of hand-crafted rules. Such rules can be formed on, for example, term co-occurrence, grammar commonalities, structure of the content etc. and allow for extracting information based on these specifications. Next, the rules are implemented by a skilled developer and tested against the dataset which was initially annotated manually by people involving either crowd-sourcing or experts. When the quality of the automatically extracted information is not satisfactory, the implemented rules are re-evaluated.

Advantages & Disadvantages

This pattern requires generally no deep technical skills for the rule-crafting person and saves time by not having to implement complicated algorithms. However the major disadvantage of the pattern is that it is hard to standardise the rules created for a specific collection to be applicable to resource sets that differ in format, structure, language, topic etc. However, the approach allows for achieving high quality when the rules have been paid enough attention and have been created and implemented correctly.

Having domain experts manually annotate the dataset can be more costly and can be an inefficient use of their time, since this is often a task which doesn't require specific domain knowledge. Therefore the task can often be performed by a non-expert unless required otherwise. In cases when specific knowledge is necessary, crowdsourcing can help by outlining certain requirements for the pool of people, such as specific education, or experience in the field for X number of years. Naturally, this will increase the costs, but will also provide a greater likelihood of an increase in quality. When no specific knowledge is required, using crowdsourcing with unspecialised pool of people can be much more effective, faster and less costly.

4.8 Pattern: SMRT LRN

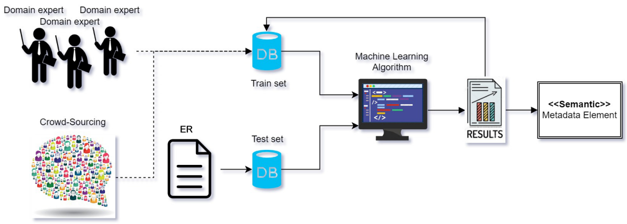


Figure 12. ***SMRT LRN*** *Pattern*

Data Types & Elements

Semantic

Motivation

Machine learning (ML) is widely utilised in a variety of different tasks, metadata extraction being one of them. Their popularity is growing beause they can be "trained" and based on known values, they can project the output of unseen data. Their popularity growth increases also the quality of the extracted information as more work is focused towards improving the performance of different ML algorithms. The current pattern (fig.12) is focused on metadata extraction with ML models. To some extent, this pattern can be considered a special case of the *SMRT D@ta* pattern, sec.(4.7). These algorithm's performance is typically evaluated with a number of metrics such as accuracy, precision, recall etc. giving the user possibility to see the impact immediately and to improve their model, if needed. This can potentially lead to really useful information being extracted and fed into the metadata of the learning resources via ML approaches such as classification, clustering, regression etc.

Requirements

There are several requirements for this pattern to work successfully. First, the algorithms can hardly be understood by everyone, therefore a professional is required who has the respective knowledge in computer science. This person will be able to select the most suitable algorithm, depending on the task and data that is to be extracted. Then, either the same expert or a developer or data scientist can implement the selected ML model. Another crucial point to consider is the fact that The supervised and reinforcement ML algorithms require *Test* and *Train* data sets. The quality of the output data depends on the quality of the train dataset. Furthermore, ML algorithms perform better with bigger sized training data. Creating this data is a tedious and time-consuming task to create. However it is required for the ML models. Some datasets exist that can be used, but if no suitable one is available, it has to be created. This is often done by crowd-sourcing in cases when no specific knowledge is required. Outsourcing to a big group of people saves time and effort. Similarly to SMRT D@TA, is specific skills are necessary, a crowd can be selected such that they have the necessary skills, which will also increase costs. Alternatively, this can be also performed by domain experts at hand. From the reliable “train” dataset, some algorithms, like reinforcement ones would ”learn” and improve.

Structural design description

First, the “train” data is generated by crowd-sourcing or domain experts. A suitable algorithm is selected for the specific task. Often, the expert or developer needs to consider pre-processing the data with techniques discussed in SMRT D@TA pattern in order to improve the performance of the algorithm. To apply the algorithm there may be other requirements such as feature selection, normalization etc. The model is trained and predicts the outcome for the required data feature. This way data values for semantic metadata elements can be predicted. At this stage, when using reinforcement ML algorithm the correct data is fed back as part of the train set to further improve the algorithm and is also fed into the metadata element.

Advantages & Disadvantages

The cost of using domain experts to create the train set for a machine learning algorithm may be high. However, it would deliver significantly higher quality, especially in case this task cannot be achieved by anyone else than experts with specific knowledge. In some situations crowd sourcing could be incorporated, but like in the previous pattern, quality would vary from domain to domain.

4.9 Pattern: Meta-walk

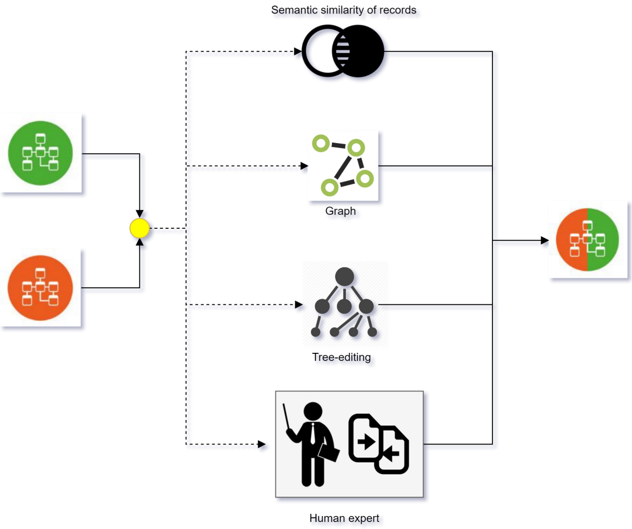


Figure 13. ***Meta-walk*** *Pattern*

Data Types & Elements

All

Motivation

Having a way to map the metadata of the learning resources in different LMS, would be very useful feature for the user who wants to annotate their own educational materials. This may save a lot of time and especially effort towards thinking about extracting solutions from scratch or via the patterns in this work. The metadata mapping issue is out of the scope of this work in general, yet, short summary of some available approaches is provided in the Meta-walk pattern (fig.13) and discussed in this subsection.

The need to map metadata from several platforms, emerges from the fact that metadata elements and values are not uniform across all educational platforms. For example, in one platform the metadata element holding the ER format can be called ”format” and in another it can be called ”MIME type”, but both contain value of ”PDF”. This makes the resources and their metadata less interoperable. Therefore, it is a good idea (sec. 3.2.1) to match schemas and extract information rather than generating new instances. However this is not a very trivial problem. Several methods are reported [45] in the literature to address it: **(1)** While it may seem like a natural and easy thing to do, often this is approached in a naive way by an expert who manually matches the schemas. This practice is infeasible, slow and inefficient, for the reason that schemas can also be very large. For the educational domain, as previously discussed, there are several standard schemas and multiple application profiles. More often the issue is approached algorithmically by **(2)** treating XML schema matching as a tree-editing problem [46] [47] and matching them bottom-up based on similarity measures. This method often involves usage of ontologies in the process of deciding which elements are e.g. parent, sibling or child of the others. **(3)** Furthermore, the matching could be looked at as a graph [48], matching elements based on proximity. **(4)** The fourth suggested approach [47] is not to compare the schemas, but the semantic similarity [49] of the metadata and thus, decide which elements match from both schemas.

Requirements

For the manual and algorithmic approaches to succeed, the schema should be available as an XML structure, and this condition is generally satisfied. Furthermore, an expert is required with certain knowledge about working with XML schemas. Manual method with an expert is most commonly performed, but requires an expert with good knowledge of the standards and types of information that should be stored in the metadata as well as common knowledge which elements may overlap and hold equal information across schemas.

Advantages & Disadvantages

Approaching the schema integration issue with the tree-editing or graph methods may introduce high cost time and budget-wise, especially if the XML schema is large. Manual approach, on the other hand could be error-prone, if the expert is not very familiar with the schema. Furthermore, manual matching can be very time consuming.

5 EXPERIMENTAL WORK

In this section a proof of concept is provided aiming towards semantic metadata extraction via the proposed design patterns in the previous section. By following the *SMRT D@Ta* (sec. 4.7) and the *SMRT LRN* (sec. 4.8) patterns, two classification approaches are applied in equal dataset. Following the provided workflow by the patterns, it is demonstrated that the patterns are high-level enough to give guidance to the user, but keep the human involvement into the in-depth decision making process on exact implementation algorithms. Therefore, one can achieve results saving himself time by not having to consider the complete solution to extraction process, but rather taking only the high-impact decisions.

5.1 Reasoning & class selection

Towards extracting semantic metadata in help of the TU Delft library project (see sec. 3.2.1), the current subsection provides reasoning about the exact metadata element to be extracted in the experiments, namely ”**Didactic intent**”.

”Didactic intent” identifies the intent of the teacher in every moment of the lecture. For instance, it marks where in the learning resource the lecturer provides a concept explanation, an example or a practical advice. Five such ”intent” classes have been selected for the algorithmic classification of the data. These classes are identified based on the pedagogical taxonomy, specified by Bloom [24] and the Six Facets of Understanding by Wiggins [50]. The choice of the classes is further based on the annotation taxonomy of Bonifazi et al. [51], which grounds on the RST structure theory [52] and justifies the usefulness of the selected classes: *Concept Description*, *Concept Mention*, *Example*, *Summary* and *Application.* Table III presents the classes, the corresponding labels, used for classification later in the experiments and the meaning of each class.

Classifying learning resources in this way demonstrates each step from the patterns and provides proof of concept regarding the main contribution of this work, namely the design patterns.

LABEL	CLASS	DESCRIPTION
CD	ConceptDecription	Explanation of the Main concept(s) of the learning resource (LR)
CM	ConceptMention	Concept or other related to the ER term mentioned but not overviewed in depth right after
SM	Summary	Summary of what has been done so far, or in this lecture or what will be done next time. Usually at the beginning and/or end of the ER.
AP	Application	Practical advise for the concept
EX	Example	Concept example. Could be of the main or sub-concept
NL	No Label	No suitable category

TABLE III. Categories for the data labelling for the "**Didactic intent**" meta element

5.2 Data collection

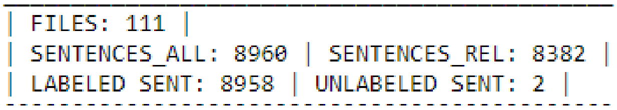
The dataset was created by crawling MOOC videos and extracting their video scripts (video subtitles). Video scripts from 11 online courses from the Coursera MOOC platform, were crawled. Under the assumption that for the purpose of the task, it is not necessary for the MOOC selection to fall under only one domain, the dataset contains video scripts from the Computer Science, Physics, Maths, Robotics and spacecraft domains. The total number of files is 556, which split by meaningful sentences, makes a considerably big dataset with 38482 sentences. Additionally, the selection of the data was based on the assumption that because the data consists of videos, and the video subtitles are extracted from these videos, the dataset practically consists of spoken language text. To that end, it is assumed that the data does not differ a lot in terms of lingual specifications, like for example, scientific papers and news articles have their own vocabulary and style of writing. Moreover, the selected classes in table III allow for this choice of courses as they are not domain specific, but rather general and aimed at the intent of the lecturer in the video, and not the topic per se.

The course choice from the Coursera MOOC platform, fell on:

- Data management
- Digital media
- Embedded operating systems
- Interactive computer graphics
- Introduction to MongoDB
- IoT Connectivity
- Logic introduction
- Mobile robots
- Particle physics
- Spacecraft dynamics
- Calculus

Considering that the data consists of spoken language, naturally it needed a careful analysis and pre-processing. The data pre-processing steps vary per pattern, however the final data was composed of a total of 38482 sentences in 556 files, ready for analysis. The detailed pre-processing steps follow in each of the pattern subsections.

Both of the patterns applied required ground truth for the evaluation and/or training steps. This ground truth is a smaller subset of manually labelled files by human experts. Due to the large amount of files and sentences, this was impossible to perform and therefore, the number of files was reduced further to one fifth of the initial amount of files, ensuring the same proportion of files as in the original course in order to provide objectivity and valid training data to the *"SMRT LRN"* algorithm. The resulting scaled-down dataset consisted of 111 video scripts with a total of 8960 sentences (fig.14). These 8960 sentences had some irrelevant sentences which were removed, thus the difference between the total and relevant sentence count. The final dataset consists of 8378 sentences, on the basis of which the final metrics were calculated.



```
| FILES: 111 |
| SENTENCES_ALL: 8960 | SENTENCES_REL: 8382 |
| LABELED SENT: 8958 | UNLABELED SENT: 2 |
---------------------------------------------
```

Figure 14. Sentence count after reduction

Finally, the reduced dataset is suitable to use for both of the patterns, followed in the next two sub-sections, namely SMRT D@TA and SMRT LRN, because SMRT D@TA applies rule-based classification which generally does not require a huge dataset, however the dataset is also big enough to train a classifier in the SMRT LRN pattern. Therefore, the dataset covers the prerequisites of both design patterns.

5.3 Methodology

This section describes in detail the decisions taken in the process of extracting semantic information by following the high-level steps from the design patterns applicable for this task. The section is a showcase of the full process involved in data extraction, by adopting the patterns, including details on condition matching for the patterns as well as in-depth details in the process of extraction itself, such as data pre-processing, algorithm selection, results, evaluation etc. The in-depth decisions, while not explicitly, are an inseparable part of the overall extraction process, covered by the patterns.

5.3.1 Application of pattern: SMRT D@ta

Selecting the correct pattern is crucial as the details of the particular implementation require time. The decision tree in fig.(5) allowed for the use of the SMRT D@TA pattern for the "**Didactic intent**" element due to the requirement of at least average quality solution, complemented by the availability of a human expert willing to spend more time than average on the analysis and implementation steps. The possibility of low budget was preferable, but not required.

The specifics of the first design pattern say that the use of the smaller subset of the MOOC data is sufficient as a huge dataset is not required. The pre-processing step is not explicitly required by the pattern, but is required to perform in general when data extraction is performed as it improves the quality of the final result. Therefore, the first step of the pattern is the techniques like natural language processing and then inference and grammar rules in order to fit the data in one of the selected classes.

As the patterns are generic, for the purpose of this experiment, the steps of the pattern were broken down into sub-steps, specific for the current experiment. They are shown in fig.(15). According to pattern, *step 1* involves the use of domain experts in the initial data analysis process. In this experiment, it was required for raw data structure analysis and decisions on the pre-processing steps. The data was first pre-processed via natural language processing techniques that are part of the NLTK package of Python, including partial stop-words removal, partial punctuation removal, sentence splitting etc. Furthermore, expressions that do not contribute contextually to the extraction process, such as mathematical formulas and numbers, were replaced with place holders like "OMITTED" and "NUMBER", so that the reader would have an idea of the context without uninformative facts. The place holders also serve to reduce the chances of errors in the classification process. Finally, all the text was made in lower-case, so that any further techniques applied would not be case-sensitive. The reason why stop-word and punctuation removal were only partial, is that certain stop words, for example "in", "to", "if", help classify the sentences. The comma was necessary for the classification of a sentence as an *"Example"*, because often the lecturer would provide a list of comma-separated values as examples.

Next, the resulting data was provided to two human annotators both with technical background. The background was not important as the labelling task was to be performed from a learner's perspective. Along with the target files, the annotators received an instructional file. The instructional file explained the structure of the textual data and based on the initial analysis by an expert, the file contained information about commonly occurring locations of specific classes, such as "Summary" which is commonly seen at the beginning or end of the script file. Moreover, the instructions provided labelled examples and terms typical for a sentence to be labelled with one of the five classes. The accuracy of this human-annotated dataset was checked by extracting a small subset of the data and giving it to an odd number of different annotators to label too. Taking the majority vote and comparing it to the originally labelled data, led to an accuracy score of the manual labelling, which is discussed in the results section (6). The final dataset of 8378 sentences was used as ground-truth for the

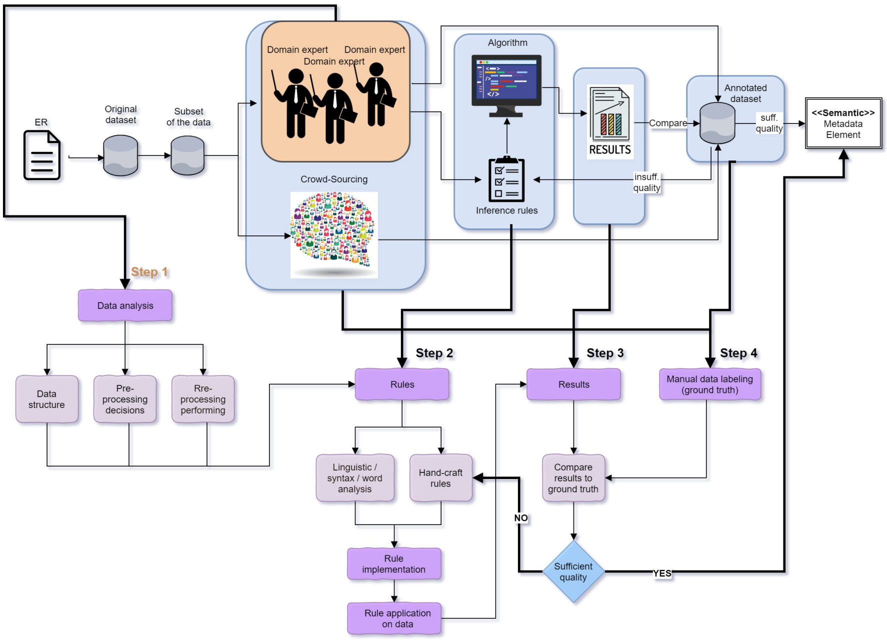


Figure 15. *Conceptual view of the **Didactic intent** extraction process with SMRT D@TA*

rule-based algorithm for classification in *step 4* of the pattern.

Next up, towards *step 2* of the pattern application, an interesting idea of facet-dictionary for scientific publications [53] was adopted and a vocabulary of *discriminative* terms, defining every class, was formalized. The identified terms were n-grams, mostly mono-, bi- and tri-grams. The list of n-grams is available in appx.{9}.

As opposed to the case of scientific papers[19], lists of specific expressions were not found online for spoken language texts. As a consequence, they had to be created manually by human experts (the same experts who labelled the ground truth data). As a result from the analysis that followed the process of vocabulary construction, several issues were identified, making this task extremely challenging:

1. The vocabulary used in spoken language differs a lot from any other kind of written language, e.g. articles, research papers, study books etc.
2. Multiple ways exist to provide the same information.
3. Some words fall into more than one class.
4. Additional ways to classify sentences alongside vocabulary are necessary in order to achieve acceptable results of 50% or more.

To elaborate further on problem #4 from the list above, an approach from previous papers [13] [14], was adapted for the needs of this extraction process. That is, a set of inference rules was defined and applied to the experiment. Like discussed in sec.(4.7), such rules draw conclusions based on pre-defined conditions. Emerging naturally from the dictionary formalization, the rules fall under *step 2* from the pattern. The rules follow logic similar to the following examples:

$$\begin{aligned} &C \leftarrow d, (T_1 \in d) \wedge \\ &\neg (TN_1 \in d \vee TN_2 \in d \vee \dots TN_n \in d) \end{aligned} \tag{1}$$

$$\begin{aligned} &C \leftarrow d, (T_1 \in d \vee \dots \vee T_n \in d) \wedge \\ &(GT_1 \in d \vee GT_2 \in d) \end{aligned} \tag{2}$$

Expression (1) classifies a document d to category C if term T_1 occurs in the document and none of the terms, $(TN_1 \dots TN_n)$ occur in the same document, where TN defines the "negative terms", e.g. the terms that must not co-exist with the main term for the current literal. For example, if a document (or in our context, a sentence), contains the tri-gram "in other words" and does not contain "should", "must" or "have to", the document is classified as *ConceptDescription* while if it does contain any of the three aforementioned words, then there's a chance that the

[19] http://www.kfs.edu.eg/com/pdf/2082015294739.pdf

sentence is, instead, an *Application*, i.e. practical advice.

An interesting finding, coming from the analysis of the data, is that sentences that fall into specific classes are usually discussed in a particular English grammar tense. To illustrate expression (2), a sentence would be classified as a *Summary* if it contains the term "summarize" or "in other words", provided that the requirement of the sentence being in either past or future grammar tense, is also satisfied. Grammar Tenses are denoted with GT in expression (2) above. One sentence can trigger more than one rule at the same time. For example, the tri-gram "in other words" triggers a rule for labelling as *Summary*, *Application* and *ConceptDescription*, however the sentence will be classified as *Application* only if the grammar tense requirement is satisfied and one extra term from a certain set of terms co-occur with the required tri-gram ("could", "should", "would" etc.).

A total of 21 inference rules were defined based on the linguistic and vocabulary analysis from step 1. The full list is available in appx.{8}. During the process of classification with the implemented rules, it was found that the algorithm misclassified some labels in particular cases, such as classifying a sentence as *ConceptMention* when human experts classify it as *Summary.* This output was not incorrect per se. As a matter of fact, it correctly identified when a concept is mentioned for the first time. Therefore, the rules were corrected as much as possible, as part of *step 3*. However some clashes caused by multi-rule triggering for the same sentence, could not be avoided. To get over this issue, it was required to have a conflict resolution strategy. This strategy consisted of class-based rule-ordering by assigning priority to every rule based on its importance in the set of all rules. For instance, if a term occurs for the first time in the summary of the lecture, priority would be given to rule labelling the sentence as *Summary*, because the overview / summary only appears at the beginning or end of the script. Then the next occurrence of the term would be considered as *ConceptMention.* That decision is based on observations that the lecturer summarizes what the video will discuss and only after that starts to explain the concept. The rest of the priorities were also assigned based on observations. At times when no rule was triggered, a default rule was implemented, assigning the same label of the previous sentence with the idea that often times a certain explanation of a concept or a summary extends to more than one sentence. The output was evaluated manually multiple times in order to improve the algorithm.

5.3.2 Application of pattern: SMRT LRN

The second experiment, adopts the SMRT LRN design pattern and splits it into smaller project-specific sub-tasks as shown in (fig.16). The originally labelled data by human annotators, was used also for experiment two due to the fact that it had enough data to train the algorithm. However when starting with the pattern, one needs to consider whether training data is available and if not - how to obtain it (*step 1* in fig.16). This could be done by crowdsourcing, or by human experts, depending on the essence of the data and the knowledge required to complete the labelling work (*step 2*). From the previous experiment, a conclusion can be drawn that in the context of this work, crowdsourcing using several people would have been the better choice. Next, similarly to experiment one, pre-processing was required for the classification task (*step 3*). Due to the nature of the classification algorithm applied, more thorough pre-processing was required. First, the text was split into appropriate sentences. Second, all stop words were removed as they would introduce bias to the algorithm. Third, all previously added "OMITTED" and "NUMBER" markers were treated as stop words and were removed. Fourth, the sentences were tokenized, i.e. split into words and then tagged with a part of speech tagger. Finally, the words were stemmed and lemmatized in order to unify all forms of a word into a single word, e.g. "gone", "going" and "went" would be mapped to "go".

When focusing on algorithm selection, the task of intent mining can be approached in several ways (as seen in relevant literature). First, as a text classification or text categorization problem, applying machine learning models. Second, as a sequential segmentation or deep learning issue [54]. Third, as a clustering issue combined with class-specific thesaurus [55]. The current work approaches it as a classification problem with a supervised machine learning model.

Having produced the training set, a probabilis-

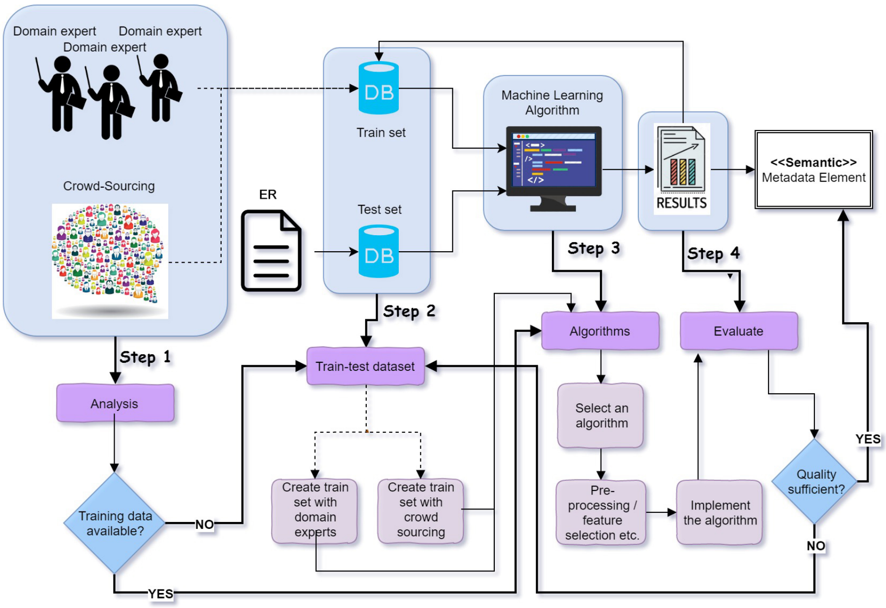


Figure 16. *Conceptual view of the* ***Didactic intent*** *extraction process with SMRT LRN*

tic Naive Bayes model is applied. Naive Bayes classifier is a "naive" model, predicting unseen data by computing probability on the basis of provided correct observations[56]. Because of the dataset essence, around 6000 sentences were labelled as *ConceptDescription*, leaving the rest of the data to be split into the other four classes, ending up with a very unbalanced dataset. The count of sentences per class for both experiments in provided in fig.(17) in the Results section. This dataset was not suitable for Naive Bayes. Therefore, 250 observations were extracted per class producing a balanced dataset allowing for the algorithm to be trained better, because it is crucial that the training data is as accurate as possible. Before training the model, Bag of Words was used to convert the words into vectors and TF-IDF to normalize them. After providing the data to the classifier, it was first split into train and test datasets in a ratio of 80 % to 20 % by randomly selecting the sentences every time. Next, the model was trained on the train set and tested on the test set. The results were averaged on the basis of running the model 10 times. The results are discussed in the next section. Unlike here, when using a supervised learning algorithm, the correctly identified values can be fed back to the train set and used to improve the quality of the algorithm. This is seen in (*step 4*) in fig.(16)

The difficulty in this approach is not algorithm-wise, but time-wise. The training phase can take a substantial amount of time depending on the dataset size. With an increasing dataset size for labelling, and that data being optionally used in next iterations to improve the algorithm, that would mean training set size increase and subsequently, will lead to a rise in time-complexity.

6 RESULTS & EVALUATION

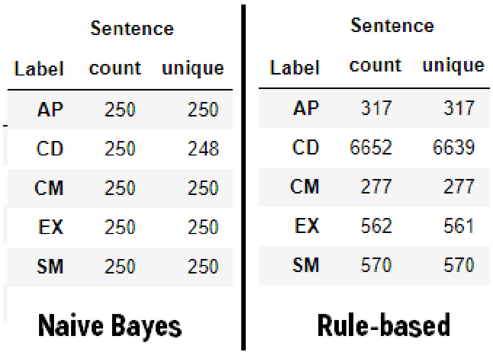
The main idea of providing design patters to practitioners is to ease their job and help them implement metadata extraction patterns easily, which, on its end would improve content findability and re-use. Quality score above 50% for the results, produced by following the proposed design patterns, could evaluate the patterns as useful. Therefore, this section discusses the pattern application results and evaluation, split by experiments, e.g.by patterns.

The core thing for one to succeed in applying the design patterns in this work, is identification of the right one to use, which depends on the requirements of the pattern. To get an idea whether the pattern identification process is intuitive, three people were asked to identify what type of data they need by giving them an example element "*Bitrate*" and asking them to find similar elements from the metadata taxonomy in fig.(1) and then to follow the decision tree on fig.(5) to a specific pattern. The taxonomy includes a big number of types of data, both from a functional point of view and from the point of view of extraction methodology, alongside example metadata element for each. It would be considered useful if a user can successfully determine the type of an example metadata element by inspecting the diagram. Once the user successfully identifies the type of the given element, they need to follow the decision tree diagram, based on their requirements for costs, output quality and time, willing to spend on the task. The people, involved into this small experiment, were able to easily identify the type of data for "*Bitrate*" and then found the correct pattern to use further. Therefore, both the taxonomy and the decision tree diagrams can be applied successfully.

After conducting experiment one (sec.5.3.1), following the SMRT D@TA pattern, the performance of the algorithm was assessed by having the respective precision, recall, f1 and accuracy metrics, shown in table (IV). Since the rule-based classification works by applying a certain set of logical rules, created by human expert, the number of sentence occurrences per class should not affect the final output. However assessing the results of the classification in the table, it was assumed that it does partially affects the output. The reason for this could be that the default rule assigns the label of the previous sentence, if no rule is triggered and since the most commonly seen label is "CD", this may explain the output. Generally, the labelled dataset of around 9000 sentences is more than enough for the human expert to draw conclusions for the logical rules, however it is hardly possible for a person to go over all of it very carefully. Normally, a much smaller size, e.g. half or even less than the current one, is enough for this purpose. Here, the big responsibility falls on the expert who will analyse the data and hand-craft the rules. The classification process that was applied on the dataset did not result in excellent performance likely due to the fact that the hand-crafted rules were implemented based on conclusions drawn from a smaller subset of the dataset, therefore it is possible that some crucial criteria for the rules, were missed. Additionally, some sentences often trigger more than one inference rule, causing the algorithm sometimes to select the incorrect one. The prioritization conflict-resolution technique was able to increase the results slightly. As seen in fig.(17), results for CD have highest scores due to the majority of the sentences being "ConceptDescription" while Application (AP) is seen rather rarely. AP was also harder to distinguish for particular courses, for example Calculus. An interesting finding was the indication of incorrectly categorized sentences in the ground truth data, as a result of human annotation being error-prone. Some of these sentences were correctly recognized by the rule-based classification approach later. This proves the importance of the manual annotation of the ground truth dataset. This fact has likely contributed to the final result metrics too. With an increase of the number of human annotators, the quality of the classification task can improve significantly.

Classifier	Class	Prec	Rec	F1	Acc
Rule-based	CM	0.57	0.18	0.27	0.97
	CD	0.90	0.72	0.80	0.71
	AP	0.11	0.30	0.16	0.88
	SM	0.30	0.44	0.35	0.89
	EX	0.21	0.50	0.30	0.84
	AVG	**0.42**	**0.43**	**0.38**	**0.86**
Naive Bayes	CM	0.97	0.77	0.86	–
	CD	0.78	0.90	0.84	–
	AP	0.87	0.83	0.85	–
	SM	0.82	0.87	0.84	–
	EX	0.82	0.85	0.83	–
	AVG	**0.85**	**0.84**	**0.84**	**0.84**

TABLE IV. Classification metrics for Rule-based vs. Naive Bayes



Naive Bayes

Label	Sentence count	Sentence unique
AP	250	250
CD	250	248
CM	250	250
EX	250	250
SM	250	250

Rule-based

Label	Sentence count	Sentence unique
AP	317	317
CD	6652	6639
CM	277	277
EX	562	561
SM	570	570

Figure 17. Label count

Since the specific classification performance serves as a proof of concept for this work, the accuracy score for the first experiment, being over 60 %, this leads us to consider the pattern SMRT D@TA applicable in situations when a practitioner has a problem of metadata extraction and does not know where to begin. Providing taxonomy and the decision tree, the user can find his way to the SMRT D@TA pattern, provided that he needs to extract semantic metadata. Applying the pattern steps by customizing them to his own specific problem, allows the user not to think for the solution from scratch.

Applying the second pattern SMRT LRN to extract the "Didactic intent" metadata element via a machine learning algorithm, shows that the pattern can provide a less time-consuming technique to the user. The core cause of potential issues with inaccurate supervised learning models, is the train dataset, such as seen in experiment (5.3.2).

To obtain a valid accuracy score of the initial labelling process, a smaller subset of the script files were selected and provided it to three new annotators, unrelated to the individuals who initially labelled the dataset. They were asked to assign labels by following the instructional file which gives information on commonly seen locations for some classes in the script files and a list of terms that could identify a sentence as a certain class, for example the occurrence of the term "summarize" is very likely to indicate a sentence that is a summary of the lecture and i mostly see at the end of the script file. Based on the output from this labelling the accuracy of the train set could be assessed, that is 83.31 %.

From fig (IV), it appears that Naive Bayes performed much better than the rule-classification in terms of the balanced f1 score. Data extraction from spoken language, however, is also challenging with rule-based classification because it is based on grammar rules and often includes slang words. It also contains a lot of "filler" words such as "well", "uhm", "okay", "right" etc. Having said that, conclusion is that for data, derived from spoken language, the pre-processing steps need to be extensive and to consider multiple points of view before applying an algorithm. Doing that could potentially improve the quality of the result in both types of extraction methods.

Nevertheless, the results from experiment two show that it is very easy to follow the steps of the pattern and end up with, in this case, a moderate-to-high-quality results in case that a suitable high quality train dataset is available at hand.

7 CONCLUSIONS & FUTURE WORK

The number of learning resources is growing with the increased interest in online education. However creating more and more resources on the same topic is not only impractical for the authors, who spend unnecessary amount of time on the task of creating content for their courses, but also poses a challenge for the learner to find high quality courses online. This work intends to contribute into solving this problem, by helping learning resources become more accessible for sharing and re-use. This can happen if the resources are accompanied by consistent and high quality metadata. This work proposed a metadata type taxonomy, helping individuals figure out the type of metadata they needs to extract. Moreover, this work contributes with several design patterns that can ease the work of the content authors towards metadata extraction or completion and thereafter, can improve content shareability and re-use. Being high-level, the proposed design patterns are easy to adapt to the needs of the individual and help them collect the necessary type of data, as classified in the proposed taxonomy in fig.(1).

As part of a showcase with the library of TU Delft, this thesis paper provides proof of concept by extracting semantic metadata for an element called "**Didactic intent**" via two of the proposed algorithms, namely SMRT D@TA and SMRT LRN. The most obvious finding to emerge from the experimental part of this work, is that in order to extract semantic metadata, one needs to consider two important aspects: (1) the ground truth or train data needs to be very accurate and (2) content analysis needs to be performed very carefully. Both of these aspects can improve or reduce the final output quality. Overall, While the metric scores of the implemented algorithms can be improved, the results shows that adopting the proposed patterns is possible and time-saving. This is also because the patterns provide a ready-solution, which eliminates the necessity for the user to think about a solution from scratch. On top of that, semantic metadata is the most challenging type of metadata to extract due to the necessity of ER content analysis, therefore assuming the rest of the patterns are also easily applicable.

While the experiments show the applicability of the design patterns with an average-to-high-quality result when extracting semantic metadata for "**Didactic intent**", the patterns can be improved further and do not claim coverage of all possible solutions.

The proposed design patterns could be improved in the future by providing commonly applied methodologies, such as specific algorithms and thus, making them more low-level and specific to the goal, for example for content personalisation. Furthermore, with respect to experiment one, the inference rules should be improved with more in-depth linguistic analysis and by identifying a way to reduce rule clashes. Additionally, the proposed patterns can be further evaluated by doing an empirical study among practitioners on real MOOC platforms and other content management systems. Moreover, with regards to the rule-based classification for spoken language texts, it would be interesting to create a multi-purpose list of commonly occurring phrases, useful for future applications and to try to improve the proposed rules in this thesis paper.

Acknowledgements

This project would have been impossible without the expertise of Dr. Christoph Lofi and prof. Dr. Geert-Jan Houben who guided me throughout the process with knowledge, useful advices and motivation.

Furthermore I'd like to thank the people who helped me label the data for the experiments, namely Filip Hristovski, Veronika Krumova and Monique Arends and my beloved father Dimitar Dimitrov.

I express my gratitude also towards Radena Lancheva, Monique Arends, and Shruti Khullar, my friends with excellent English language skills who helped me proofread my thesis report and suggested multiple useful tips on making it better.

I'd like to thank Dimitar Dimitrov and Filip Hristovski for the moral support throughout the process. Last but not least, huge thanks to my best friends who also supported me and motivated me: Ioana Leontiuc, Veronika Krumova, Desislava Popnikolova, Desislava Evstatieva, Radena Lancheva, and of course my favourite cousin Boryana Borisova!

References

[1] Nikolaos Palavitsinis, Nikos Manouselis, and Salvador Sanchez-Alonso. **Metadata quality in learning object repositories: a case study**. *The Electronic Library*, 32(1):62–82, 2014.

[2] M. Derntl. ***Patterns for Person Centered E-learning***. Dissertationen zu Datenbanken und Informationssystemen. Aka, 2005.

[3] Erich Gamma, Richard Helm, Ralph E. Johnson, and John M. Vlissides. **Design Patterns: Abstraction and Reuse of Object-Oriented Design**. In *Proceedings of the 7th European Conference on Object-Oriented Programming*, ECOOP '93, pages 406–431, Berlin, Heidelberg, 1993. Springer-Verlag.

[4] Cory Doctorow. **Metacrap: Putting the torch to seven straw-men of the meta-utopia**. `https://people.well.com/user/doctorow/metacrap.htm`, 26 August 2001.

[5] IMS Global Learning Consortium. **IEEE LOM specification.** `http://ltsc.ieee.org`.

[6] IMS global learning consortium. **Learning Resource Meta-data Specification**. `https://www.imsglobal.org/metadata/index.html`.

[7] E. Duval, E. Vervaet, B. Verhoeven, K. Hendrikx, K. Cardinaels, H. OliviC, E. Forte, F. Haenni, K. Warkentyne, M. Wentland Forte, and F. Simillion. **Managing Digital Educational Resources with the ARIADNE Metadata System**. In *2014 International Conference on Intelligent Networking and Collaborative Systems*, pages 145–171, 06 Mar 2009.

[8] Kathryn Eccles and Andrew Greg. **Your Paintings Tagger: Crowdsourcing descriptive metadata for a national virtual collection**. *Crowdsourcing our Cultural Heritage*, pages 185–208, 2014.

[9] Kris Cardinaels, Michael Meire, and Erik Duval. **Automating Metadata Generation: The Simple Indexing Interface**. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 548–556, New York, NY, USA, 2005. ACM.

[10] Maria Elisabete Catarino and Ana Alice Baptista. **Relating Folksonomies with Dublin Core**. *Int. J. Metadata Semant. Ontologies*, 5(4):285–295, September 2010.

[11] M. Lama, J. C. Vidal, E. O. Garcia, A. Bugarin, and S. Barro. **Semantic Linking of a Learning Object Repository to DBpedia**. In *2011 IEEE 11th International Conference on Advanced Learning Technologies*, pages 460–464, July 2011.

[12] O. Motelet and N. Baloian. **Hybrid System for Generating Learning Object Metadata**. In *Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06)*, pages 563–567, July 2006.

[13] Marek Hatala and Griff Richards. **"Value-Added Metatagging: Ontology and Rule Based Methods for Smarter Metadata"**. In Michael Schröder and Gerd Wagner, editors, *Rules and Rule Markup Languages for the Semantic Web*, pages 65–80, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[14] P. Rullo, V. L. Policicchio, C. Cumbo, and S. Iiritano. **Olex: Effective Rule Learning for Text Categorization**. *IEEE Transactions on Knowledge and Data Engineering*, 21(8):1118–1132, Aug 2009.

[15] R. Farhat and B. Jebali. **OBSemE: An ontology-based semantic metadata extraction system for learning objects**. In *Fourth International Conference on Information and Communication Technology and Accessibility (ICTA)*, pages 1–5, Oct 2013.

[16] E. H. Othman, S. Abdelali, and E. B. Jaber. **Education data mining: Mining MOOCs videos using metadata based approach**. In *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pages 531–534, Oct 2016.

[17] F. A. Dorça, V. C. Carvalho, M. M. Mendes, R. D. Araújo, H. N. Ferreira, and R. G. Cattelan. **An Approach for Automatic and Dynamic Analysis of Learning Objects Repositories through Ontologies and Data Mining Techniques for Supporting Personalized Recommendation of Content in Adaptive and Intelligent Educational Systems**. In *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)*, pages 514–516, July 2017.

[18] N. Capuano, A. Gaeta, F. Orciuoli, and S. Paolozzi. **Exploiting Tagging in Ontology-based e-Learning**.

[19] A. M. Pradhan and A. S. Varde. **Ontology based meta knowledge extraction with Semantic Web tools for ubiquitous computing**. In *2016 IEEE 7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pages 1–6, Oct 2016.

[20] Darrell Porcello and Sherry Hsi. **Crowdsourcing and Curating Online Education Resources**. *Science*, 341(6143):240–241, 2013.

[21] Jung-ran Park and Andrew Brenza. **Evaluation of Semi-Automatic Metadata Generation Tools: A Survey of the Current State of the Art**. 2015.

[22] Nicola Capuano, Sergio Miranda, and Francesco Orciuoli. **IWT: A Semantic Web-based Educational System**, 12 2009.

[23] S. Miranda and P. Ritrovato. **Automatic Extraction of Metadata from Learning Objects**. In *2014 International Conference on Intelligent Networking and Collaborative Systems*, pages 704–709, Sept 2014.

[24] M. Forehand. **Bloom's taxonomy: Original and revised**. *Emerging perspectives on learning, teaching, and technology*, 8, 2005.

[25] Gaetano Bruno Ronsivalle, Simona Carta, and Vanessa Metus. ***L'Arte della progettazione didattica. Dall'analisi dei contenuti alla valutazione dell'efficacia***. 01 2009.

[26] E. H. Othman, S. Abdelali, and E. B. Jaber. **Education data mining: Mining MOOCs videos using metadata based approach**. In *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pages 531–534, Oct 2016.

[27] Coffield, F., Moseley, D., Hall, E. and Ecclestone, K. **Learning styles and pedagogy in post-16 learning: a systematic and critical review**, 2009.

[28] Harry Halpin, Valentin Robu, and Hana Shepherd. **The Complex Dynamics of Collaborative Tagging**. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 211–220, New York, NY, USA, 2007. ACM.

[29] S. Ternier, K. Verbert, G. Parra, B. Vandeputte, J. Klerkx, E. Duval, V. Ordonez, and X. Ochoa. **The Ariadne Infrastructure for Managing and Storing Metadata**. *IEEE Internet Computing*, 13(4):18–25, July 2009.

[30] Martin Fowler. ***Analysis Patterns: Reusable Objects Models***. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

[31] Marian Simko and Maria Bielikova. **Discovering hierarchical relationships in educational content**. In *Advances in Web-Based Learning -ICWL 2012*, page 132–141, 2012.

[32] John Atkinson, Andrea Gonzalez, Mauricio Munoz, and Hernan Astudillo. **Web Metadata Extraction and Semantic Indexing for Learning Objects Extraction**. *Applied Intelligence*, 41(2):649–664, September 2014.

[33] Jung-ran Park and Andrew Brenza. ***Semi-automatic Metadata Generation Workflow for Developing a Continuing Education Resource Repository***, pages 235–245. Springer International Publishing, Cham, 2015.

[34] Devshri Roy, Sudeshna Sarkar, and Sujoy Ghose. **Automatic Extraction of Pedagogic Metadata from Learning Content**. *Int. J. Artif. Intell. Ed.*, 18(2):97–118, April 2008.

[35] Jie Tang, Mingcai Hong, Duo Zhang, Bangyong Liang, and Juanzi Li. Information extraction: Methodologies and applications.

[36] Olivier Motelet and Nelson A. Baloian.

[37] Zachary A. Pardos, Steven Tang, Daniel Davis, and Christopher Vu Le. **Enabling Real-Time Adaptivity in MOOCs with a Personalized Next-Step Recommendation Framework**. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, L@S '17, pages 23–32, New York, NY, USA, 2017. ACM.

[38] Xavier Ochoa, Kris Cardinaels, Michael Meire, and Erik Duval. **Frameworks for the Automatic Indexation of Learning Management Systems Content into Learning Object Repositories**. 2005.

[39] Srimathi H. **Knowledge Representation of LMS using Ontology**. 6, 09 2010.

[40] P. Vrablecová and M. Šimko. **Supporting Semantic Annotation of Educational Content by Automatic Extraction of Hierarchical Domain Relationships**. *IEEE Transactions on Learning Technologies*, 9(3):285–298, July 2016.

[41] W. Guo and D. Chen. **Semantic Approach for e-learning System**. In *Computer and Computational Sciences, 2006. IMSCCS '06. First International Multi-Symposiums on*, volume 2, pages 442–446, June 2006.

[42] E. M. Kalogeraki, C. Troussas, D. Apostolou, M. Virvou, and T. Panayiotopoulos. **Ontology-based model for learning object metadata**. In *2016 7th International Conference on Information, Intelligence, Systems Applications (IISA)*, pages 1–6, July 2016.

[43] Jennifer Wortman Vaughan. **Incentives and the Crowd**. *XRDS*, 24(1):42–46, September 2017.

[44] Dharitri Misra, Siyuan Chen, and George R Thoma. **A System for Automated Extraction of Metadata from Scanned Documents using Layout Recognition and String Pattern Search Models**. In *IS and T's Archiving Conference*, pages 107–12, 2009.

[45] S. Amir, I. M. Bilasco, T. Danisman, T. Urruty, I. Elsayad, and C. Djeraba. **Schema matching for integrating multimedia metadata**. In *2010 International Conference on Machine and Web Intelligence*, pages 234–239, Oct 2010.

[46] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. **Generic Schema Matching with Cupid**. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 49–58, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[47] Elisa Bertino, Giovanna Guerrini, and Marco Mesiti. **Measuring the structural similarity among XML documents and DTDs**. *Journal of Intelligent Information Systems*, 30(1):55–92, Feb 2008.

[48] S. Melnik, H. Garcia-Molina, and E. Rahm. **Similarity flooding: a versatile graph matching algorithm and its application to schema matching**. In *Proceedings 18th International Conference on Data Engineering*, pages 117–128, 2002.

[49] Aida Boukottaya and Christine Vanoirbeek. **Schema Matching for Transforming Structured Documents**. In *Proceedings of the 2005 ACM Symposium on Document Engineering*, DocEng '05, pages 101–110, New York, NY, USA, 2005. ACM.

[50] Grant P. Wiggins, Jay. McTighe, and Hawker Brownlow Education. ***Understanding by design / Grant Wiggins, Jay McTighe***. Hawker Brownlow Education Moorabbin, Vic, 2nd expanded ed. edition, 2005.

[51] F Bonifazi, S Levialdi, Paola Rizzo, and R Trinchese. A web-based annotation tool supporting e-learning. pages 123–128, 01 2002.

[52] William C Mann and Sandra Thompson. Rhetorical structure theory: A theory of text organization, 01 1987.

[53] Sepideh Mesbah, Kyriakos Fragkeskos, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben. Semantic annotation of data processing pipelines in scientific publications. In Eva Blomqvist, Diana Maynard, Aldo Gangemi, Rinke Hoekstra, Pascal Hitzler, and Olaf Hartig, editors, *The Semantic Web*, pages 321–336, Cham, 2017. Springer International Publishing.

[54] T. L. Luong, M. S. Cao, D. T. Le, and X. H. Phan. Intent extraction from social media texts using sequential segmentation and deep learning models. In *2017 9th International Conference on Knowledge and Systems Engineering (KSE)*, pages 215–220, Oct 2017.

[55] S. Repp and M. Meinel. **Semantic indexing for recorded educational lecture videos**. In *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*, pages 5 pp.–245, March 2006.

[56] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, March 2002.

Appendices

A LOM Standard

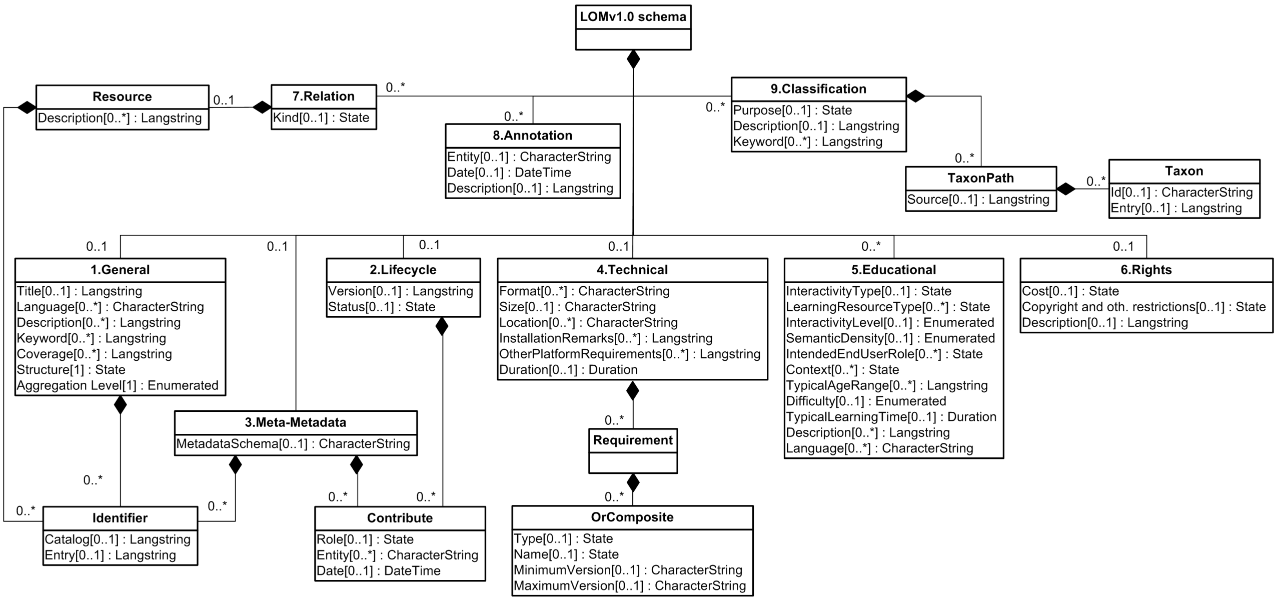

LOMv1.0 schema
Resource
Description[0..*] : Langstring
0..1
7.Relation
Kind[0..1] : State
0..*
0..*
8.Annotation
Entity[0..1] : CharacterString
Date[0..1] : DateTime
Description[0..1] : Langstring
0..*
9.Classification
Purpose[0..1] : State
Description[0..1] : Langstring
Keyword[0..*] : Langstring
0..*
TaxonPath
Source[0..1] : Langstring
0..*
Taxon
Id[0..1] : CharacterString
Entry[0..1] : Langstring
0..1
0..1
0..1
0..1
0..*
0..1
1.General
Title[0..1] : Langstring
Language[0..*] : CharacterString
Description[0..*] : Langstring
Keyword[0..*] : Langstring
Coverage[0..*] : Langstring
Structure[1] : State
Aggregation Level[1] : Enumerated
2.Lifecycle
Version[0..1] : Langstring
Status[0..1] : State
4.Technical
Format[0..*] : CharacterString
Size[0..1] : CharacterString
Location[0..*] : CharacterString
InstallationRemarks[0..*] : Langstring
OtherPlatformRequirements[0..*] : Langstring
Duration[0..1] : Duration
5.Educational
InteractivityType[0..1] : State
LearningResourceType[0..*] : State
InteractivityLevel[0..1] : Enumerated
SemanticDensity[0..1] : Enumerated
IntendedEndUserRole[0..*] : State
Context[0..*] : State
TypicalAgeRange[0..*] : Langstring
Difficulty[0..1] : Enumerated
TypicalLearningTime[0..1] : Duration
Description[0..*] : Langstring
Language[0..*] : CharacterString
6.Rights
Cost[0..1] : State
Copyright and oth. restrictions[0..1] : State
Description[0..1] : Langstring
3.Meta-Metadata
MetadataSchema[0..1] : CharacterString
0..*
Requirement
0..*
0..*
0..*
0..*
0..*
0..*
Identifier
Catalog[0..1] : Langstring
Entry[0..1] : Langstring
Contribute
Role[0..1] : State
Entity[0..*] : CharacterString
Date[0..1] : DateTime
OrComposite
Type[0..1] : State
Name[0..1] : State
MinimumVersion[0..1] : CharacterString
MaximumVersion[0..1] : CharacterString

Figure 18. LOM standard element set

B LOM Educational Class

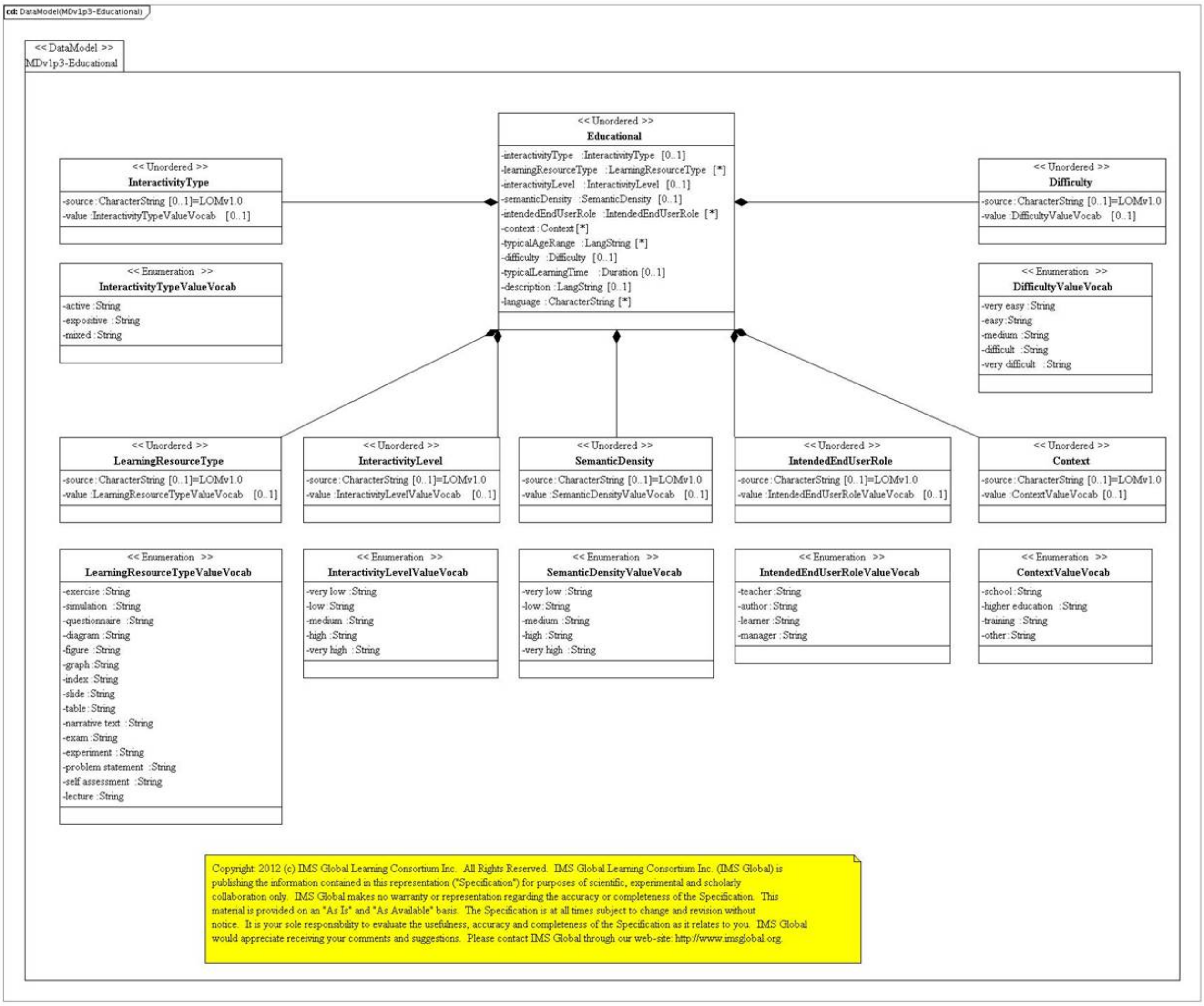

cd: DataModel(MDv1p3-Educational)
<< DataModel >>
MDv1p3-Educational
<< Unordered >>
Educational
-interactivityType :InteractivityType [0..1]
-learningResourceType :LearningResourceType [*]
-interactivityLevel :InteractivityLevel [0..1]
-semanticDensity :SemanticDensity [0..1]
-intendedEndUserRole :IntendedEndUserRole [*]
-context : Context [*]
-typicalAgeRange :LangString [*]
-difficulty :Difficulty [0..1]
-typicalLearningTime :Duration [0..1]
-description :LangString [0..1]
-language :CharacterString [*]
<< Unordered >>
InteractivityType
-source : CharacterString [0..1]=LOMv1.0
-value :InteractivityTypeValueVocab [0..1]
<< Enumeration >>
InteractivityTypeValueVocab
-active :String
-expositive :String
-mixed :String
<< Unordered >>
Difficulty
-source : CharacterString [0..1]=LOMv1.0
-value :DifficultyValueVocab [0..1]
<< Enumeration >>
DifficultyValueVocab
-very easy :String
-easy :String
-medium :String
-difficult :String
-very difficult :String
<< Unordered >>
LearningResourceType
-source : CharacterString [0..1]=LOMv1.0
-value :LearningResourceTypeValueVocab [0..1]
<< Unordered >>
InteractivityLevel
-source : CharacterString [0..1]=LOMv1.0
-value :InteractivityLevelValueVocab [0..1]
<< Unordered >>
SemanticDensity
-source : CharacterString [0..1]=LOMv1.0
-value :SemanticDensityValueVocab [0..1]
<< Unordered >>
IntendedEndUserRole
-source : CharacterString [0..1]=LOMv1.0
-value :IntendedEndUserRoleValueVocab [0..1]
<< Unordered >>
Context
-source : CharacterString [0..1]=LOMv1.0
-value :ContextValueVocab [0..1]
<< Enumeration >>
LearningResourceTypeValueVocab
-exercise :String
-simulation :String
-questionnaire :String
-diagram :String
-figure :String
-graph :String
-index :String
-slide :String
-table :String
-narrative text :String
-exam :String
-experiment :String
-problem statement :String
-self assessment :String
-lecture :String
<< Enumeration >>
InteractivityLevelValueVocab
-very low :String
-low :String
-medium :String
-high :String
-very high :String
<< Enumeration >>
SemanticDensityValueVocab
-very low :String
-low :String
-medium :String
-high :String
-very high :String
<< Enumeration >>
IntendedEndUserRoleValueVocab
-teacher :String
-author :String
-learner :String
-manager :String
<< Enumeration >>
ContextValueVocab
-school :String
-higher education :String
-training :String
-other :String

Figure 19. LOM Educational class element set and per-element vocabulary values

C Pattern Decision Tree

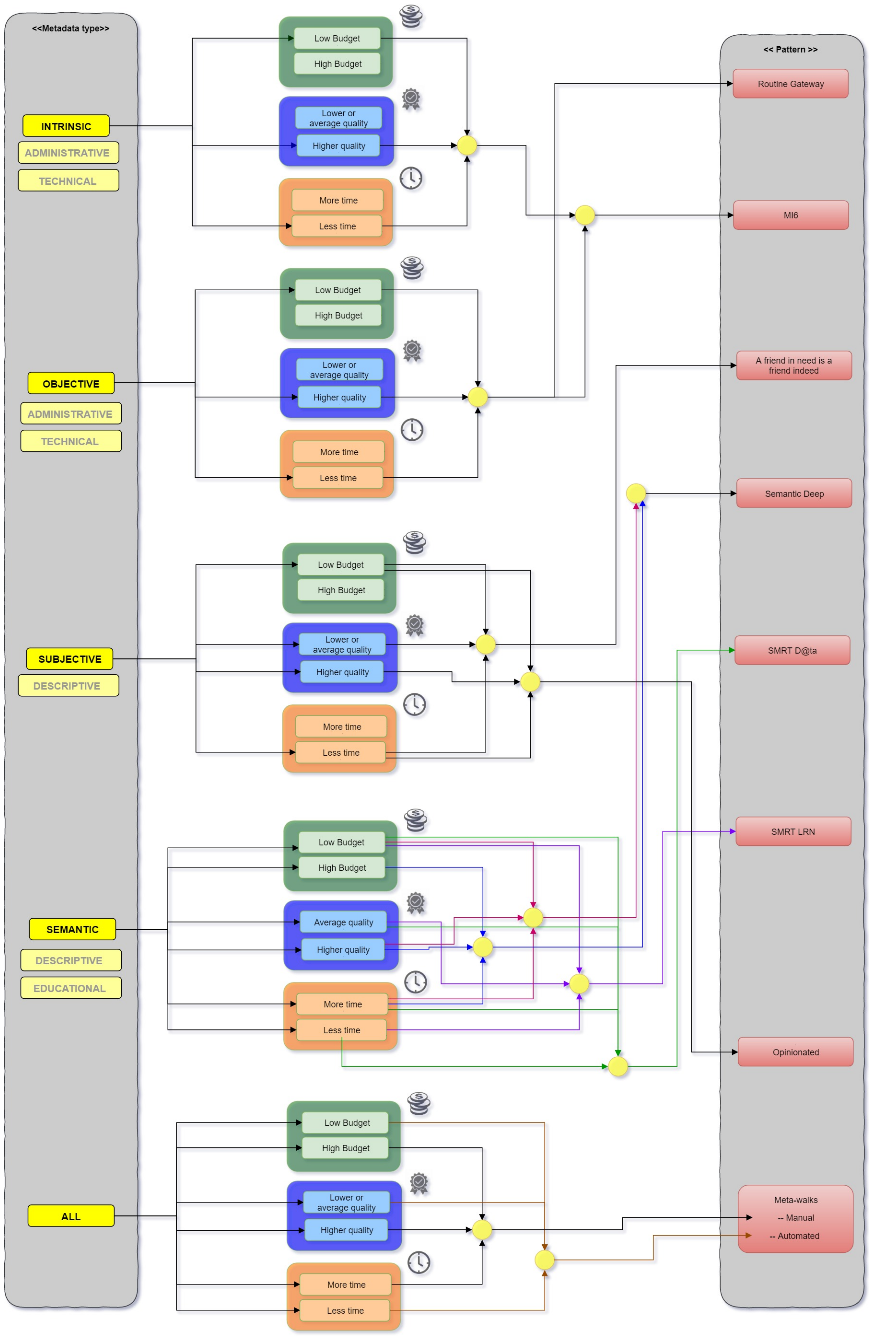
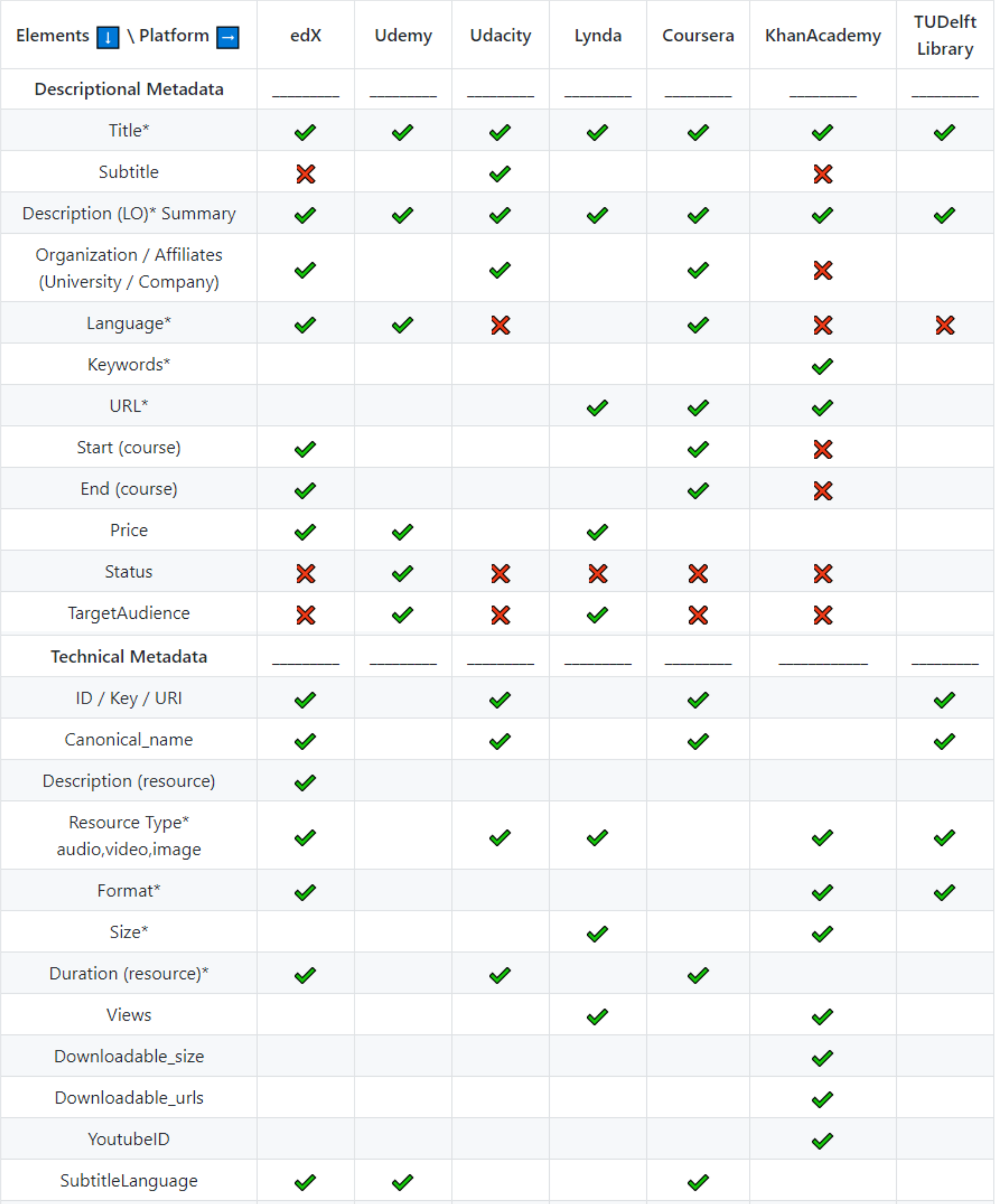


Figure 20. Decision tree on pattern use

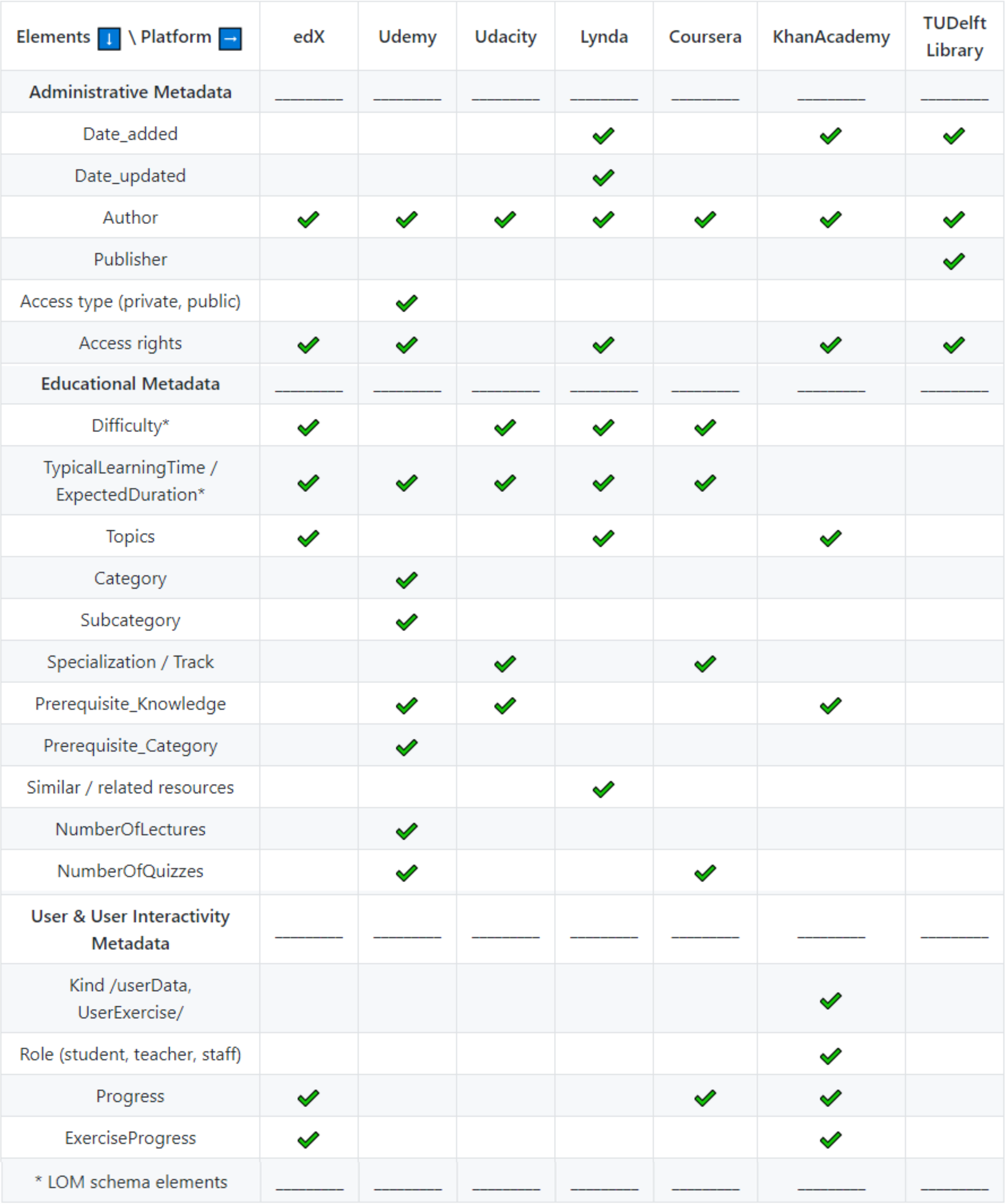
D Metadata element comparison in MOOC platforms part 1



Elements ↓ \ Platform →	edX	Udemy	Udacity	Lynda	Coursera	KhanAcademy	TUDelft Library
Descriptional Metadata	_______	_______	_______	_______	_______	_______	_______
Title*	✓	✓	✓	✓	✓	✓	✓
Subtitle	✗		✓			✗	
Description (LO)* Summary	✓	✓	✓	✓	✓	✓	✓
Organization / Affiliates (University / Company)	✓		✓		✓	✗	
Language*	✓	✓	✗		✓	✗	✗
Keywords*						✓	
URL*				✓	✓	✓	
Start (course)	✓				✓	✗	
End (course)	✓				✓	✗	
Price	✓	✓		✓			
Status	✗	✓	✗	✗	✗	✗	
TargetAudience	✗	✓	✗	✓	✗	✗	
Technical Metadata	_______	_______	_______	_______	_______	_________	_______
ID / Key / URI	✓		✓		✓		✓
Canonical_name	✓		✓		✓		✓
Description (resource)	✓						
Resource Type* audio,video,image	✓		✓	✓		✓	✓
Format*	✓					✓	✓
Size*				✓		✓	
Duration (resource)*	✓		✓		✓		
Views				✓		✓	
Downloadable_size						✓	
Downloadable_urls						✓	
YoutubeID						✓	
SubtitleLanguage	✓	✓			✓		

Figure 21. MOOC Platform comparison Part 1

E Metadata element comparison in MOOC platforms part 2



Elements ↓ \ Platform →	edX	Udemy	Udacity	Lynda	Coursera	KhanAcademy	TUDelft Library
Administrative Metadata	________	________	________	________	________	________	________
Date_added				✓		✓	✓
Date_updated				✓			
Author	✓	✓	✓	✓	✓	✓	✓
Publisher							✓
Access type (private, public)		✓					
Access rights	✓	✓		✓		✓	✓
Educational Metadata	________	________	________	________	________	________	________
Difficulty*	✓		✓	✓	✓		
TypicalLearningTime / ExpectedDuration*	✓	✓	✓	✓	✓		
Topics	✓			✓		✓	
Category		✓					
Subcategory		✓					
Specialization / Track			✓		✓		
Prerequisite_Knowledge		✓	✓			✓	
Prerequisite_Category		✓					
Similar / related resources				✓			
NumberOfLectures		✓					
NumberOfQuizzes		✓			✓		
User & User Interactivity Metadata	________	________	________	________	________	________	________
Kind /userData, UserExercise/						✓	
Role (student, teacher, staff)						✓	
Progress	✓				✓	✓	
ExerciseProgress	✓					✓	
* LOM schema elements	________	________	________	________	________	________	________

Figure 22. MOOC Platform comparison Part 2

F MOOC Metadata comparison Statistics

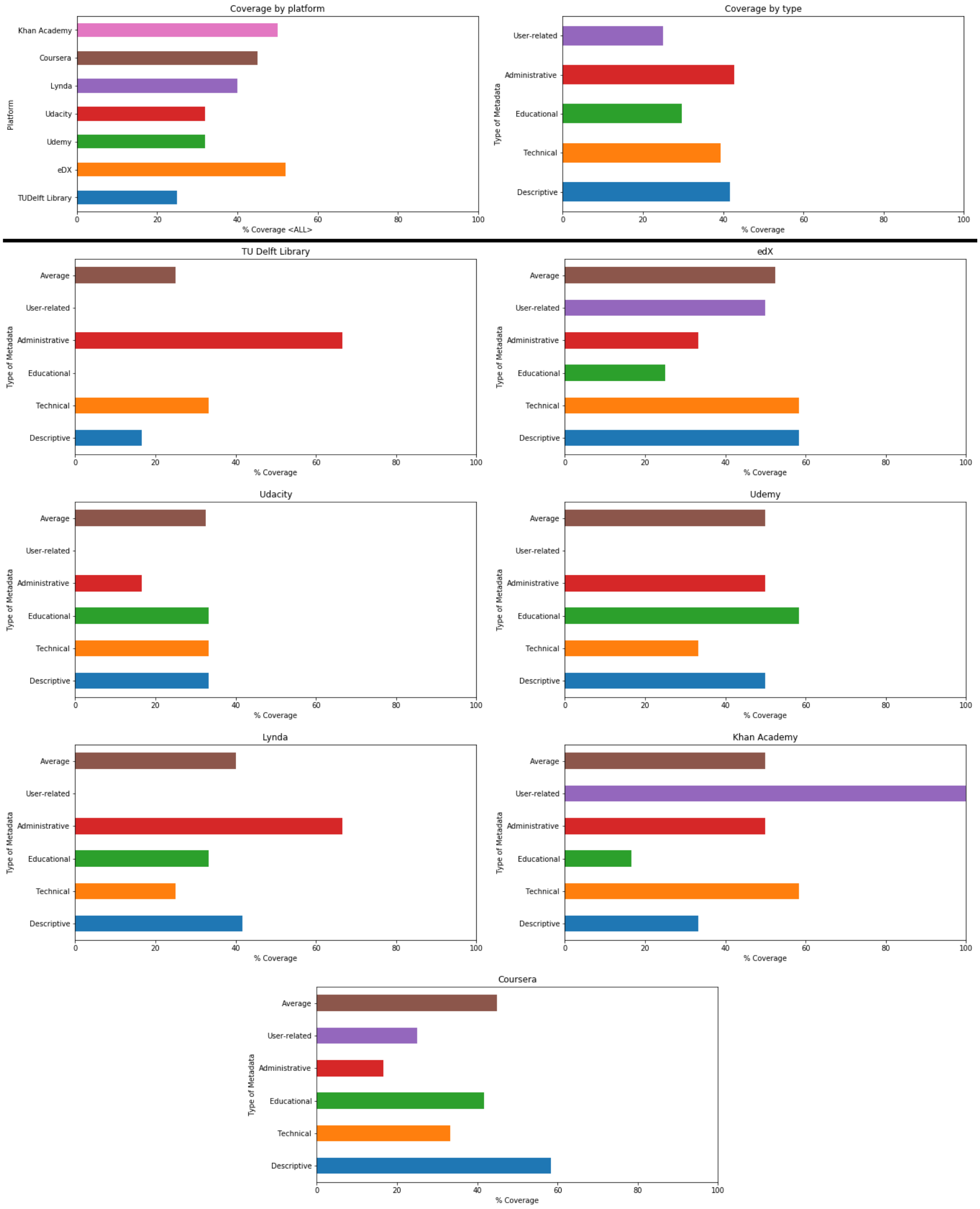
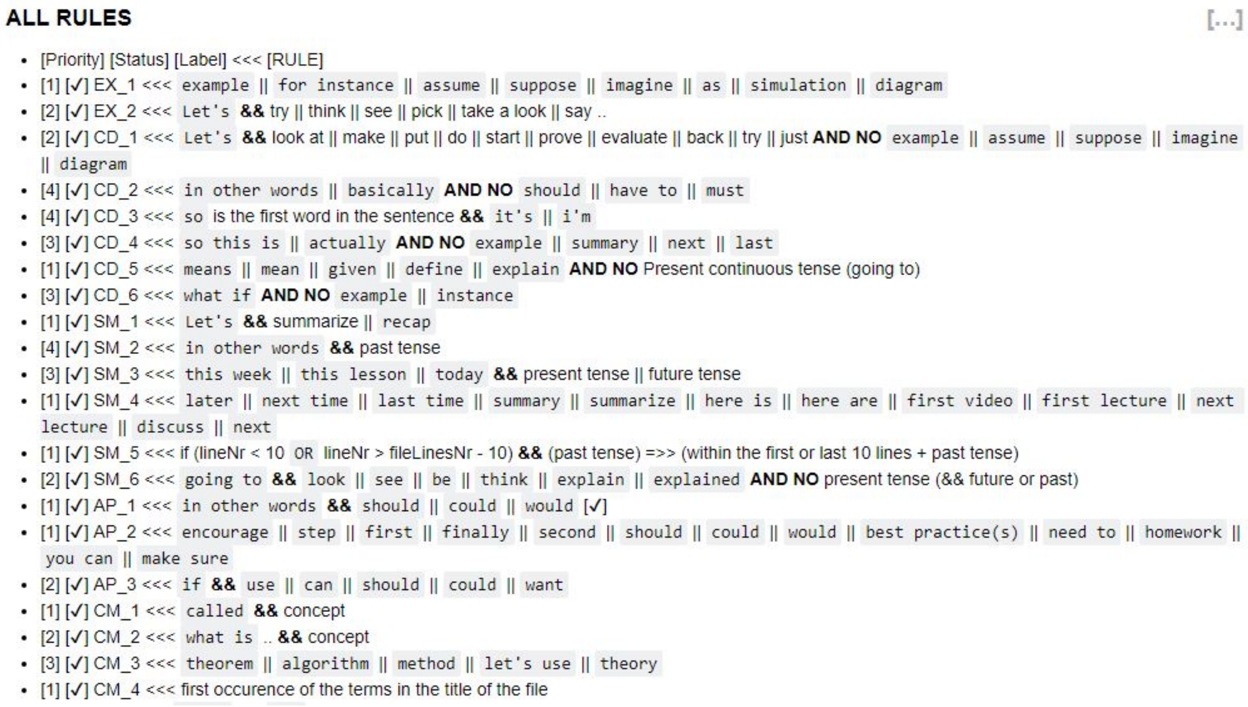


Figure 23. MOOC Metadata comparison Statistics

8 Inference rules



ALL RULES [...]

- [Priority] [Status] [Label] <<< [RULE]
- [1] [✓] EX_1 <<< `example` || `for instance` || `assume` || `suppose` || `imagine` || `as` || `simulation` || `diagram`
- [2] [✓] EX_2 <<< `Let's` **&&** try || think || see || pick || take a look || say ..
- [2] [✓] CD_1 <<< `Let's` **&&** look at || make || put || do || start || prove || evaluate || back || try || just **AND NO** `example` || `assume` || `suppose` || `imagine` || `diagram`
- [4] [✓] CD_2 <<< `in other words` || `basically` **AND NO** `should` || `have to` || `must`
- [4] [✓] CD_3 <<< `so` is the first word in the sentence **&&** `it's` || `i'm`
- [3] [✓] CD_4 <<< `so this is` || `actually` **AND NO** `example` || `summary` || `next` || `last`
- [1] [✓] CD_5 <<< `means` || `mean` || `given` || `define` || `explain` **AND NO** Present continuous tense (going to)
- [3] [✓] CD_6 <<< `what if` **AND NO** `example` || `instance`
- [1] [✓] SM_1 <<< `Let's` **&&** summarize || `recap`
- [4] [✓] SM_2 <<< `in other words` **&&** past tense
- [3] [✓] SM_3 <<< `this week` || `this lesson` || `today` **&&** present tense || future tense
- [1] [✓] SM_4 <<< `later` || `next time` || `last time` || `summary` || `summarize` || `here is` || `here are` || `first video` || `first lecture` || `next lecture` || `discuss` || `next`
- [1] [✓] SM_5 <<< if (lineNr < 10 `OR` lineNr > fileLinesNr - 10) **&&** (past tense) =>> (within the first or last 10 lines + past tense)
- [2] [✓] SM_6 <<< `going to` **&&** `look` || `see` || `be` || `think` || `explain` || `explained` **AND NO** present tense (&& future or past)
- [1] [✓] AP_1 <<< `in other words` **&&** `should` || `could` || `would` [✓]
- [1] [✓] AP_2 <<< `encourage` || `step` || `first` || `finally` || `second` || `should` || `could` || `would` || `best practice(s)` || `need to` || `homework` || `you can` || `make sure`
- [2] [✓] AP_3 <<< `if` **&&** `use` || `can` || `should` || `could` || `want`
- [1] [✓] CM_1 <<< `called` **&&** concept
- [2] [✓] CM_2 <<< `what is` .. **&&** concept
- [3] [✓] CM_3 <<< `theorem` || `algorithm` || `method` || `let's use` || `theory`
- [1] [✓] CM_4 <<< first occurence of the terms in the title of the file

Figure 24. Inference rules implemented and tested

9 Class vocabularies

```
conceptdescr_dict = ['this research','refers to','last but not least','number one', '
    number two','number three','drawback','advantages', 'disadvantages','benefits','
    drawbacks','defined', 'there are many types','important to understand', 'let\'s
    look at','when we use the term','imagine that', 'refers to as','key concepts','
    this is a term','more specifically', 'helps to','is used to','makes it easier to',
     'provide information about','when talking about','critical to understand', 'let\'
    s take a look at','crucial','can be used to','ensures', 'rule of thumb','consider'
    ,'last but not least','topic of this', 'topic of today','one way to','another
    possibility is', 'next one is','problem','to address','want to do','different ways
    ', 'approach is to','tries to','try to','topic is','method','task', 'in this why',
    'we specify','by combining','results','because','due to', 'in other words','in
    fact','means','done through','implies','if you take']

conceptmention_dict = ['called','use the term','more specifically','this includes', '
    activities surrounding','show you','additionally','should include', 'types of','
    recommends','known as','known for','advantages to', 'disadvantages to','benefits',
    'drawbacks','plays a key role', 'refers to as','first one is','we introduce','
    techniques', 'second one is','traditional','method','-based','what\'s', 'what is',
     'says']

application_dict = ['best practices','best practice','means that','practical benefit',
     'sorts of things','good practices','best to','you should','you shouldn\'t', 'may
    find that','can use','this is important','particularly importnant', 'is why','
    useful to','recommended that','consider','useful to','instead', 'tips','tip','
    advice','advised','encourage','experiment','explore','remember']

example_dict = ['example','there are many','many types','include','can see','hands on'
    , 'includes','exemplar','prototype','sample','case','illustration','analogy', 'let
    \'s think of','may want to','some of','most of','different types of', 'e.g.','for
    example','key aspects','key concepts','in other words', 'of these are','of them
    are','rather than','such as','pros and cons', 'a long list','various','for
    instance','it\'s like','different ways', 'tried to','instead of','say','challenge'
    ,'assume']

summary_dict = ['by the end of','as you know','have a good understanding','will begin
    by', 'will cover','well delve','will discuss','will be able to','will understand',
     'to summarize','summary','take away','to conclude','as you know','introduced', '
    be aware','we\'ve looked at','last week we discussed','discuss how to', 'will
    describe','the goal','know that','step','first','second','third', 'finally','next'
    ,'consideting','can see','we\'ve seen','learned','recall', 'remember','last time']
```

Listing 1. vocabulary terms per class