# Enriching RDF-based Document Management System with Semantic-based Reasoning

Maria Assunta Cappelli Ashley Caselli Giovanna Di Marzo Serugendo
CUI – Centre Universitaire d'Informatique
Université de Genève, Geneva, Switzerland
E-mail: maria.cappelli@unige.ch

## Abstract

*Paper-dependent companies spend most of their time organising and searching for documents, such as invoices, contracts, and budgets. To carry out those tasks, fiduciaries, insurance brokers, and other companies are pioneering Document Management Systems (DMSs). However, there is currently no DMS that allows the classification, understanding, and reasoning of a bundle of documents delivered by customers, and that helps companies create customer profiles to make better and faster decisions. Our proposal aims at easing the tasks of companies dealing with administrative documents of different kinds. We propose a semantic rule-based approach that permits to recognise and classify customers' documents, as well as to reason over those documents and create customer profiles based on the extracted information. We provide and discuss a case study grounded on the Swiss tax declaration. We developed a full Swiss tax ontology composed of 241 classes, as well as 120 semantic reasoning rules fully validated on the minimum set of administrative documents necessary to fill the Swiss tax declarations. Our approach automates activities related to the management of administrative documents, the profile of their clients, and the administrative documents they must deliver.*

***Index terms—*** Automatic Document Processing, Data Management Systems, Knowledge Graph-based Approach, Reasoning Engine

## 1. Introduction

Metadata-driven document management platforms have drastically simplified the work of document-dependent companies in the past decade. They enable professionals to find the right information, automate business processes, and enforce information control in any environment. Efficient management of documents is crucial not only for the optimal finding and use of documents but also for an effective and efficient work organisation. Gorelashvili et al. [5] note that in the legal sector, automated document management is essential to improve and streamline the way lawyers manage their practice. Document Management Systems (DMSs) ensure that documents are easily accessible, well-organised, and protected. Abbassova et al. [1] share the same opinion. They highlight the beneficial effects of DMSs on workflow forms, by automating the routing of documents between people, eliminating bottlenecks, and optimising business processes. They highlight the benefits of DMSs on the workflow assuming that the DMSs ensure more accurate organisation of business processes within the company through effective management and support the quality system in line with international norms, as well as efficient storage, management, and access to information and knowledge. However, the DMSs structure raises issues regarding the quality and completeness of the information sought, as the information is processed according to metadata. DMSs solutions either have considerable activation processes – and the associated fees – or are not well-suited for industry-specific professions. Also, since most solutions are business-based, there are currently no solutions for automating access to critical documents for individual consumers. There is not yet a DMS that allows the classification, understanding, and reasoning of customers' documents, automatically processing a bundle of customers' documents and creating a customer profile, in compliance to regulations. Tax declarations or insurance brokerage-related documents are still manually processed, transferred by e-mail or via consumers' cloud platforms.

This paper proposes a semantic rule-based approach for RDF-based DMSs. We designed a rule-based process that dynamically builds, reasons, and takes into account users' profiles and underlying regulations. This process is based on the information extracted from the documents users provide. Based on ontology capturing Swiss tax declaration, we designed rules on which the SHACL-based reasoner runs to derive inferences from the asserted RDF triples of

DOI reference number - 10.18293/DMSVIVA23-034

various tax households. A more detailed description of the approach can be found in the technical report [3].

The remainder of the paper is organised as follows. Section 2 provides an overview of existing approaches related to the DMSs. Section 3 describes the proposed approach and a case study. Section 4 shows our rule-based methodology. The SHACL-based implementation of the rules and their execution is shown in section 5. Finally, section 6 provides the evaluation of the defined rules, and Section 7 concludes the paper.

## 2. Related work

To handle the variety of documents written in different formats within an automated process, some works propose a semantic management approach for heterogeneous documents through the use of ontologies. They formalise the structure and interrelations of individual document types. These approaches monitor the process, take care of the various dependencies between documents, analyse the consequences of the changes made in one document on other documents, and engineer the synchronisation steps necessary to obtain a consistent document collection.

Motta et al. [8] propose an ontology-driven approach to enrich documents. This approach enables the development and integration of formal knowledge models with archives of documents. It extends what is currently available using "standard" information retrieval and search facilities by providing intelligent knowledge retrieval and additional knowledge-intensive services.

Fuertes et al. [4] developed an ontology for DMS concerning the construction field. The ontology aims at classifying documents along the life cycle of the research project, decreasing the interoperability and information exchange issues, establishing a hierarchical structure of the different areas that conform to the lifecycle of such projects, and finally enabling an interrelated system between these areas.

Doc2KG is a framework that provides a continuous conversion of open data to a knowledge graph, exploiting the existing domain ontology standards. The system handles the initial conversion of a DMS to a knowledge graph and supports the perpetual population of the created knowledge graph with new documents. The authors rely on a combination of NLP techniques to facilitate the information extraction and on constraint-solving techniques for knowledge graph creation and manipulation [12].

The Semantic Document Management System (SDMS) leverages a semantic approach for managing the lifecycle of semantic documents, from their authoring and publication to archival. The system allows defining documents as composite resources with document content units that are uniquely identified and semantically annotated. One relevant feature of the SDMS is the ability to share and exchange the document contents and semantics by the users [10].

Several other research deal with an SDMS and some of them cover relevant aspects of document modelling. However, none of them relies on semantic rules. For example, Yen-Hsien Lee et al. [7] develop a domain-specific ontology to support automatic document categorisation. The ontology includes a complete and detailed hierarchy of concepts that are used to represent documents related to information systems and technology as a set of concepts with relative weights. While scholars recognise the advantages of using an ontology with classes in terms of interpretability and understandability of classification decisions, no reference is made to the definition of semantic rules to make the use of the ontology more flexible. Sheng et al. [11] propose the use of ontology in the context of e-governance to model government data and create a semantic environment for managing government information. They present a semantic-based e-government system structure and use OWL as an ontology description language, which aims to provide the basis for data sharing and analysis. The authors detail the conceptual entity, conceptual property, and relationship between concepts that correspond respectively with class, property, and axioms in the OWL language. However, they do not model semantic rules to define constraints and restrictions on the data, ensuring that they are consistent with the ontological structure they have defined, or to exploit their reasoning power.

## 3. General approach

We propose a semantic rule-based approach for helping companies process (e.g. administrative) documents for their customers. Our proposal addresses the following research questions:

A) How to classify and multi-label a document based on extracted information providing its key features?

B) How to build and update clients' profiles based on the documents provided and the information extracted from those documents?

C) How can a reasoning process determine which documents the customers must deliver based on their profiles?

### 3.1. Case study

In this paper, we focus on the households' Swiss tax declaration, and the documents required to fill the tax declarations. We limit our case study to private households profiles composed of a single person, a widow/er, couples, households with or without children or any other dependent people, retired, or working. We also limited our

case study to the minimum set of administrative documents necessary to fill the Swiss tax declarations of the households described above, namely: yearly revenue, bank statements, health insurance policies and benefits, and family allowances, concerning every household member. We have included the health insurance statements as they are mandatory in Switzerland and anybody must provide them for tax declaration purposes.

We consider a scenario of a tax household consisting of two working parents with children. As each parent is employed, data is extracted from their two salary certificates. The data extraction process identifies the main features of the document that are necessary conditions for a document to be classified as a salary certificate. In response to the first research question, rules are applied to classify the documents as salary certificates based on the extracted features. The system then assigns a double tag of "*Tax*" and "*Income*" to the salary certificates. In response to the second research question, the system profiles the two parents as employees. Finally, in relation to the third research question, the system identifies other necessary documents that the two parents and their children must provide, such as health insurance.

### 3.2. Global workflow and architecture

Figure 1 depicts the global overview of the workflow of our approach, whose detailed description can be found in [3]. Such an architecture is composed of three modules: *(i)* actual documents (native PDFs or scanned documents) are processed through a *Document classification and information extraction module*. This module generates *JSON files for each document*, identifying its class (e.g., health insurance policy), as well as specific information extracted from the document (e.g., date, amount); *(ii)* assuming that the documents are identified as being part of the bundle of documents belonging to a specific tax household, the information extracted from the documents is also used to feed *JSON files profiles* of the household and its various members (e.g., widow/er, child, etc.); *(iii)* JSON information is then mapped to RDF by the *Reasoning, Labelling and Profiles updates* module, using an ontology for Swiss tax declaration, as well as people profiles. This module also contains a semantic rule-based reasoner, which serves on the one hand to update the profiles' information (e.g., health insurance policy for a new child means that child must be added to the household, possibly changing the household profile from couple without children to couple with children), and on the other hand to identify any missing document, based on the existing profiles, of the household (e.g., health policy or benefits are missing for a person identified as being a part of the household).

The details of the *Reasoning, Labelling, and Profiles update* module are the followings: *(i)* an *ontology of the Swiss tax* declaration terms and documents, based on actual official tax declaration legal documents and on actual documents needed to fill the tax declaration. The ontology defines concepts such as tax household, tax sections, documents, people, as well as profiles; *(ii)* the *rules*, defined for documents validation, updating the profiles based on new information, identifying missing documents (e.g. not provided in the bundle), and labelling documents; and *(iii)* the *RDF data* mapped from the JSON files (actual data) that contain information extracted from the documents using an information extraction process.

## 4. Rule-Based Methodology

We develop our semantically enriched DMS by designing semantic rules addressing the points A), B), and C) of Section 3.

**Classification and multi-labelling rules.** One or more label is assigned to each document to allow automated organisation of the documents into several predefined categories. Table 1 shows a *Salary Certificate* document as being multi-labelled as both a *Tax* and an *Income* document.

**Customer profile rules.** Infer the users' profile by analysing the documents they provided (*Document* → *User Profile*). We refer to them as **direct rules**. Table 1 shows that if a user delivers a *Salary Certificate*, then the user is tagged as being an *Employee*.

| **DOCUMENT ⟶ LABEL** | | |
|---|---|---|
| *Salary Certificate* | tagged as | *Tax* |
| *Salary Certificate* | tagged as | *Income* |
| **DOCUMENT ⟶ USER PROFILE** | | |
| *User* | delivers *Salary Certificate* | *User* is an *Employee* |
| *User* | delivers *Health Insurance Policy* | *User* is an *Person* |
| **USER PROFILE ⟶ DOCUMENT** | | |
| *Employee* | has to deliver | *Salary Certificate* |
| *Person* | has to deliver | *Health Insurance Policy* |

Table 1: Examples of the defined rules

**Documents delivery rules.** Infer which documents match the profile of the clients (*User Profile* → *Document*). We qualify these rules as **inverse rules**. Table 1 shows that if a *Person* is tagged as *Employee*, then he/she must deliver a *Salary Certificate*. Additionally, any *Person* must deliver a *Health Insurance Policy* document.

## 5. Implementation

We implemented the aforementioned rules as inference rules using the SHACL language [6] – a W3C standard developed by the RDF Data Shapes Working Group for RDF graph validation[1]. It is a highly expressive language that lets

[1]RDF Data Shapes Working Group Charter, 2017, `https://www.w3.org/groups/wg/data-shapes/charters`

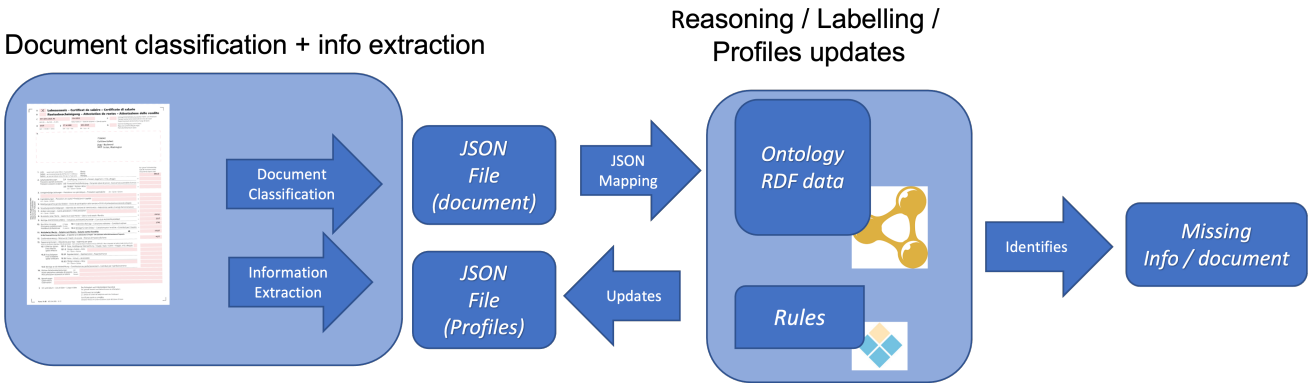


Figure 1: Overview of the global workflow as presented in [3]

its users write conditions, called shapes, that an RDF data graph must satisfy. In addition to the validation features, it also provides constructs for writing data transformations and inference rules through the SHACL Advanced Features vocabulary [2].

We created an ontology using Protégé [9] to represent the Swiss tax vocabulary. The ontology creation has been supported by extracting the domain's terms from the tax guide 2020 of the Geneva canton. Such an ontology defines a common vocabulary that will be used throughout the process to describe the *documents* and *user profiles*. As the system will be used in connection with tax declarations, tax items are also represented. Therefore, it defines classes for representing administrative and tax documents (e.g., salary certificates, bank account statements, health insurance, family allowances, etc...), and the users' profiles (e.g., married, single, in cohabitation, widow/er, divorced, separated, employee, self-employed, etc...). The ontology is composed of 241 classes, 24 data type properties, 615 axioms, and 15 object properties. Further information on the ontology can be found in [3].

After defining the ontology, we stated 92 property shapes and three sets of semantic rules (discussed in Section 3) resulting in a total of 120 rules, which included 78 multi-label rules, 21 customer profile rules, and 21 document delivery rules [3].

◇ FIRST TASK: A) *Classification of documents*

For each document a user may deliver, we identify its *sine qua non* elements. For instance, a *Salary Certificate* must include the following elements containing information about the person: employee's surname and first name, employee's address, employer, net and gross salary, etc... We then define a SHACL shape for each document type with the relevant elements the document must contain. Each element is represented as a SHACL property shape. For instance, as shown in Listing 1, a *Salary Certificate* document must contain: the employee's surname (`impots:PersonSurnameShape`) and first name (`impots:PersonFirstNameShape`), one and only one *Employer* (property named *"EmployerPropertyShape"*), and the amount (`impots:AmountShape`). By using such a SHACL shape, we can run a validation process that validates (or does not) the document as of the corresponding type. This can also be interpreted as "if the document contains all the *sine qua non* elements, namely the validation is positive, then it belongs to the specific class" and thus is assigned with that class.

```
impots:SalaryCertificateShape
  rdf:type sh:NodeShape ;
  sh:property [
      sh:path impots:employer ;
      sh:minCount 1 ;
      sh:maxCount 1 ;
      sh:class impots:Employer ;
      sh:name "EmployerPropertyShape" ;
    ] ;
  sh:property impots:AmountShape ;
  sh:property impots:PersonSurnameShape ;
  sh:property impots:PersonFirstNameShape ;
  sh:targetClass impots:SalaryCertificate ;
.
```

Listing 1: Relevant features of a *Salary Certificate* document represented as SHACL shapes

◇ SECOND TASK: A) *Multi-labelling documents*

Multi-labelling rules assign one (or more) label to each document. By using the assigned labels, the documents can then be automatically organised into predefined categories. We define such rules as SHACL inference rules. Their execution generates inferred triples of the form:

`< document impots:tag label >`

where *document* is the RDF individual of the document that is being labelled; impots:tag is a data property, defined in the ontology, for assigning the label to a document; and *label* is a string literal (xsd:string) containing the actual text value of the label. Listing 2 shows a rule labelling a document of type SalaryCertificate as both a tax document ("*Tax*" label) and an income document ("*Income*" label).

```
impots:SalaryCertificateShape
    rdf:type sh:NodeShape ;
    sh:rule [
        rdf:type sh:TripleRule ;
        sh:subject sh:this ;
        sh:predicate impots:tag ;
        sh:object "Tax" ;
    ] ;
    sh:rule [
        rdf:type sh:TripleRule ;
        sh:subject sh:this ;
        sh:predicate impots:tag ;
        sh:object "Income" ;
    ] ;
    sh:targetClass impots:SalaryCertificate ;
.
```

Listing 2: SHACL inference rule for labelling a document of type *SalaryCertificate*

In case there exist any triples in the data graph that fulfil the following conditions: *(i)* there exists a document (:documentX) in the data graph, and *(ii)* such document is of type *SalaryCertificate* (:documentX rdf:type impots:SalaryCertificate), therefore the execution of the rules shown in Listing 2 infers new triples shown in Listing 3.

```
:documentX impots:tag "Tax" .
:documentX impots:tag "Income" .
```

Listing 3: Triples inferred by the inference rule shown in Listing 2

◇ THIRD TASK: B) *Customer profile rules*

As the multi-labelling rules, the customers' profile rules are defined as SHACL inference rules. Listing 4 shows an example of such rules. Contrary to the example previously shown, where the targeted documents were all the RDF individuals of a defined class, this example shows an extended targeting condition expressed using the SPARQL language.

```
impots:SalaryCertificate_Employee-Shape
    rdf:type sh:NodeShape ;
    sh:rule [
        rdf:type sh:TripleRule ;
        sh:object impots:Employee ;
        sh:predicate rdf:type ;
        sh:subject sh:this ;
    ] ;
    sh:target [
        rdf:type sh:SPARQLTarget ;
        sh:prefixes impots: rdf: ;
        sh:select """
            SELECT ?this
            WHERE {
                ?sc rdf:type
                    impots:SalaryCertificate .
                ?sc impots:recipient ?this .
                ?this rdf:type impots:Person .
            }
        """ ;
    ] ;
.
```

Listing 4: SHACL direct rule inferring the *Employee* profile of a *Person* from the provided *SalaryCertificate*

The execution of the direct rule defined in Listing 4 infers new triples of the form:

```
< person rdf:type impots:Employee >
```

where *person* corresponds to the specific RDF individual; rdf:type is the property used to state that a resource is an instance of a class; and impots:Employee is the inferred class to which *person* belongs.

◇ FOURTH TASK: C) *Documents delivery rules*

Listings 5 and 6 show examples of user profile rules.

```
impots:EmployeeShape
  rdf:type sh:NodeShape ;
  sh:rule [
      rdf:type sh:TripleRule ;
      sh:subject sh:this ;
      sh:predicate impots:delivers ;
      sh:object impots:SalaryCertificate ;
    ] ;
  sh:targetClass impots:Employee ;
.
```

Listing 5: SHACL inverse rule inferring the need for an *Employee* profile to deliver a *SalaryCertificate*

The execution of the rule defined in Listing 5 infers new triples of the form:

```
< employee impots:delivers
impots:SalaryCertificate >
```

which means that any *Employee* must deliver a *SalaryCertificate*. The execution of the rule defined in Listing 6 infers new triples of the form:

```
impots:PersonShape
  rdf:type sh:NodeShape ;
  sh:rule [
      rdf:type sh:TripleRule ;
      sh:subject sh:this ;
      sh:predicate impots:delivers ;
      sh:object impots:HealthInsurance ;
    ] ;
  sh:targetClass impots:Person;
.
```

Listing 6: SHACL inverse rule for inferring that a *Person* profile needs to deliver a *Health Insurance*

```
< person impots:delivers
impots:HealthInsurance >
```

which in turn means that any *Person* must deliver a *HealthInsurance* policy document.

## 6. Evaluation

Concerning the evaluation of our approach we define two aspects:

- ◇ the performance of the information extraction process, which is addressed by precision, recall, and accuracy;
- ◇ the validation of the reasoning rules, which identifies missing documents or updates profiles in all cases.

Our use cases and tests are limited to a small set of documents, and the rules strongly depend on the quality of the information extraction process. Our aim with this work is to provide a proof of concept for the semantic-based approach, assuming the information is properly extracted from the documents. A more complete solution would need a larger dataset in order to be able to test all potential rules, as well as a thorough evaluation of the information extraction component.

Our focus is on the second aspect of the evaluation. Therefore, we validated all the rules regarding the points A), B) and C). Our technical report [3] provides a complete discussion of the rules validation. For testing purposes, we created synthetic data for various households and added them to a local RDF graph loaded into TopBraid Composer[2]. We used those data for testing the inference rules presented in Listing 2, 4, 5, and 6.

### 6.1. Evaluating multi-labeling rules

As we can see in Figure 2, the execution of the rules shown in Listing 2 infers two new triples that assign the two labels "*Income*" and "*Tax*" to the individual `impots:SalaryC12.3.334` which is of type *SalaryCertificate*.

| | | |
|---|---|---|
| impots:SalaryCertificate12.3.334 | impots:tag | Income |
| impots:SalaryCertificate12.3.334 | impots:tag | Tax |

Figure 2: Inferred triples that assign two labels to a document of type *SalaryCertificate*

### 6.2. Evaluating users profile rules

We defined two individuals `impots:ZolaGiovanna` and `impots:Ladoumegue_Jules`. We assume that `impots:ZolaGiovanna` delivered a *SalaryCertificate* document. Based on the direct rule defined in Listing 4, since `impots:ZolaGiovanna` delivered such a certificate, the rule infers she is an *Employee*. Figure 3 shows the inferred triples.

| | | |
|---|---|---|
| impots:ZolaGiovanna | rdf:type | impots:Employee |

Figure 3: The result of the execution of direct rules on an individual that delivered a document of type *SalaryCertificate*

Conversely, we defined `impots:Ladoumegue_Jules` as an *Employee*. Therefore, according to the inverse rule defined in Listing 5, the execution infers that since he is of class *Employee*, he must deliver a *SalaryCertificate* document. According to Listing 6, since he is also a *Person*, he needs to provide a *HealthInsurance* policy document. Figure 4 shows the mentioned inferences.

| | | |
|---|---|---|
| impots:Ladoumegue_Jules | impots:delivers | impots:SalaryCertificate |
| impots:Ladoumegue_Jules | impots:delivers | impots:HealthInsurance |

Figure 4: The result of the execution of inverse rules on an *Person* with an *Employee* profile

## 7. Conclusion and future work

In this paper, we have presented a semantic rule-based approach for a semantically enriched DMS. The adoption of such an approach would ease performing tasks such as administrative document management, user profiling, and profiling-related ones. As depicted by Figure 1, the work presented in this article is a module that may be integrated into a wider solution.

In summary, we implemented a proof of concept based on the SHACL language as well as a functional prototype

[2] `https://www.topquadrant.com/`

that is rather complete on the reasoning, labelling, and profiling module. The implementation still needs to be validated on a large dataset of documents. We focused on the validation of the semantic-based reasoning rules.

Future work will take into account the dynamicity of the profiles (a person's profile might change over time) as well as the integration of such a module into a wider DMS service.

## Acknowledgments

This research was supported by Innosuisse within the framework of the innovation project 50606.1 IP-ICT "Addmin". The authors thank Anne-Françoise Cutting-Decelle, Assane Wade, Claudine Métral, Gilles Falquet, Graham Cutting, and Sami Ghadfi for their valuable collaboration with the "Addmin" project.

## References

[1] V. Abbasova. Main concepts of the document management system required for its implementation in enterprises. *ScienceRise*, 1:32–37, 02 2020.

[2] D. Allemang, S. Steyskal, and H. Knublauch. SHACL advanced features. W3C note, W3C, June 2017.

[3] G. Di Marzo Serugendo, G. Falquet, C. Metral, M. A. Cappelli, A. Wade, S. Ghadfi, A.-F. Cutting-Decelle, A. Caselli, and G. Cutting. Addmin: Private computing for consumers' online documents access: Scientific technical report. 2022.

[4] A. Fuertes, N. Forcada, M. Casals, M. Gangolells, and X. Roca. Development of an ontology for the document management systems for construction. In *Complex Systems Concurrent Engineering*, pages 529–536. Springer, 2007.

[5] L. Gorelashvili. The importance of digitalization of legal documents preparing process and its impact on peoples' legal guarantees. In R. Geibel and S. Machavariani, editors, *Digital Management in Covid-19 Pandemic and Post-Pandemic Times*. Springer, Cham, 2023.

[6] H. Knublauch and D. Kontokostas. Shapes constraint language (SHACL). W3C recommendation, W3C, July 2017.

[7] Y.-H. Lee, P. J.-H. Hu, W.-J. Tsao, and L. Li. Use of a domain-specific ontology to support automated document categorization at the concept level: Method development and evaluation. *Expert Systems with Applications*, 174:114681, 2021.

[8] E. Motta, S. B. Shum, and J. Domingue. Ontology-driven document enrichment: principles, tools and applications. *Int. J. Hum. Comput. Stud.*, 52(6):1071–1109, 2000.

[9] M. A. Musen. The protégé project: a look back and a look forward. *AI Matters*, 1(4):4–12, 2015.

[10] S. Nesic, D. Gasevic, and M. Jazayeri. Semantic document management for collaborative learning object authoring. In *2008 Eighth IEEE International Conference on Advanced Learning Technologies*, pages 751–755. IEEE, 2008.

[11] L. Sheng and L. Lingling. Application of ontology in e-government. In *2011 Fifth International Conference on Management of e-Commerce and e-Government*, pages 93–96. IEEE, 2011.

[12] N. Stylianou, D. Vlachava, I. Konstantinidis, N. Bassiliades, and V. Peristeras. Doc2kg: Transforming document repositories to knowledge graphs. *Int. J. Semantic Web Inf. Syst.*, 18(1):1–20, 2022.