

DEPARTMENT OF
CLASSICAL PHILOLOGY AND ITALIAN STUDIES - FICLIT

SECOND CYCLE DEGREE PROGRAMME IN
DIGITAL HUMANITIES AND DIGITAL KNOWLEDGE

**TRIPLE ONTOLOGY: A REUSABLE
FORMALISATION OF SCIENTIFIC ENTITIES IN THE
SOCIAL SCIENCES AND HUMANITIES DOMAIN**

Dissertation in
85411 Programming and Data Structures (I.C.) (LM)

Supervisor

Prof. Silvio Peroni

Defended by

Alessandro Bertozzi

Co-Supervisor

Luca De Santis, Net7 Srl

Graduation Session 07/2024

Academic Year 2023/2024

Quod solus sit quod est, et qui est
(*Anselmo d'Aosta*, Proslogion)

Table of Content

Table of Content.....	3
Table of Figures.....	6
List of Tables.....	7
Table of Listings.....	8
Abstract.....	9
Chapter 1: Introduction.....	11
Chapter 2: State of the Art.....	15
2.1 Data Modelling: a definition.....	15
2.2 Data Modelling in Data Aggregator.....	16
2.2.1 ISIDORE.....	17
2.2.2 Europeana.....	19
2.2.3 OpenAlex.....	20
2.2.4 OpenCitations.....	21
2.2.5 OpenAIRE.....	24
2.3 Data Modelling in the Semantic Web.....	26
2.3.1 The Semantic Web.....	26
2.3.2 Ontology: a definition.....	26
2.3.3 Ontologies for modelling research artefacts.....	29
2.3.3.1 Semantic Publishing and Referencing (SPAR) ontologies.....	29
2.3.3.2 Research Object Create (RO-Crate).....	31
2.3.3.3 Data Catalog Vocabulary (DCAT).....	33
2.3.4 Linked Data.....	35
Chapter 3: Methodology.....	39
3.1 Preliminary Analysis.....	39
3.1.1 The case of GoTriple.....	39
3.1.1.1 GoTriple Platform Architecture.....	40
3.1.1.2 Data Collection and Normalisation Strategies.....	40
3.1.1.3 Data Model and Ontology.....	41
3.1.2 Report on Data Enrichment.....	41
3.1.3 Triple Data Model.....	42
3.2 Output of the Analysis.....	47
3.3 Ontology Design and Development.....	53
3.3.1 Simplified Agile Methodology for Ontology.....	54
3.3.1.1 Development.....	54
3.3.1.2 Reused Model.....	55
3.3.1.3 Tools.....	56
Chapter 4: Triple ontology.....	58
4.1 GoTriple Ontology Structure & Components.....	58
4.1.1 Namespaces.....	59
4.1.2 FRBR alignment with FaBiO.....	61
4.1.3 Model entities.....	63

4.1.4 Model relationships.....	67
4.1.5 SKOS.....	68
4.1.6 FOAF.....	70
4.1.7 PRO.....	71
4.1.8 Datacite.....	72
4.1.9 Schema.org.....	73
4.1.10 TVC.....	74
4.1.11 Textual Data.....	74
4.2 Application Scenario.....	76
4.2.1 Relations between Document expressions and manifestations.....	77
4.2.2 Relations Between Documents, Roles, Profiles, Online Accounts, and Agents.....	79
4.2.3 Connecting Documents to Controlled Vocabularies.....	81
Chapter 5: Development.....	83
5.1 Scenario 1: Documents.....	83
5.1.1 Motivating Scenario.....	83
5.1.2 Competency questions.....	84
5.1.3 Description.....	86
5.2 Scenario 2: Controlled Vocabulary.....	88
5.2.1 Motivating Scenario.....	88
5.2.2 Competency Questions.....	89
5.2.3 Description.....	91
5.3 Scenario 3: Roles.....	92
5.3.1 Motivating Scenario.....	92
5.3.2 Competency Questions.....	93
5.3.3 Description.....	95
5.4 Scenario 4: Subjects Coverage.....	96
5.4.1 Motivating Scenario.....	96
5.4.2 Competency Question.....	96
Figure 7: iteration 4.....	97
5.4.3 Description.....	98
5.5 Scenario 5: Discarded Entities and Documents Clustering.....	99
5.5.1 Motivating Scenario.....	99
5.5.2 Competency Question.....	99
5.5.3 Description.....	101
5.6 Scenario 6: Profile and User Account.....	102
5.6.1 Motivating Scenario.....	102
5.6.2 Competency Question.....	102
5.6.3 Description.....	104
5.7 Scenario 7: Project.....	105
5.7.1 Motivating Scenario.....	105
5.7.2 Competency Question.....	105
5.7.3 Description.....	107
Chapter 6: Discussion and conclusions.....	108

6.1 Discussion and Results.....	108
6.2 Limitations.....	112
6.3 Further Development.....	112
6.4 Conclusions.....	113
References.....	115
Bibliography.....	115
Sitography.....	122

Table of Figures

Figure 1. Application Scenario 1.....	76
Figure 2. Application Scenario 2.....	79
Figure 3. Application Scenario 3.....	80
Figure 4. Iteration 1.....	84
Figure 5. Iteration 2.....	89
Figure 6. Iteration 3.....	93
Figure 7. Iteration 4.....	96
Figure 8. Iteration 5.....	99
Figure 9. Iteration 6.....	102
Figure 10. Iteration 7.....	105

List of Tables

Table 1. Triple data model.....	42
Table 2. data model of Triple profiles.....	43
Table 3. data model of Triple projects.....	43
Table 4. Alignment of the Triple data model with OWL.....	48
Table 5. Alignment of the Triple Profile data model with OWL.....	48
Table 6. Alignment of the Triple Project data model with OWL.....	48
Table 7. Categories Triple data model fields.....	49
Table 8. Categories Triple profile fields.....	49
Table 9. Categories Triple project fields.....	50
Table 10. Triple Ontology prefix.....	58
Table 11. Triple Ontology Namespaces.....	60
Table 12. Legend of table 13.....	108
Table 13. Data Aggregator comparison: good practice coverage.....	109
Table 12. Data Aggregator comparison: entities and properties reused.....	110

Table of Listings

Listing 1. Example of a resource described in RDF linked to a resource described on dbpedia.....	35
Listing 2. RDF application scenario 1.	76
Listing 3. SPARQL query application scenario 1.....	77
Listing 4. RDF application scenario 2.....	78
Listing 5. SPARQL query application scenario 2.....	79
Listing 6. RDF application scenario 3.	80
Listing 7. SPARQL query application scenario 3.....	81

Abstract

The TRIPLE project, launched in October 2019 and coordinated by the French National Center for Scientific Research (CNRS), involved 22 partners from 15 European countries. Its primary aim was to develop the GoTriple discovery platform, a multilingual access point for discovering and reusing research artefacts in the social sciences and humanities (SSH). This thesis presented a comprehensive ontology designed to formalise the TRIPLE data model using semantic technologies. The ontology addressed the challenge of managing heterogeneous data aggregated by the Core Pipeline, which integrated research artefacts from diverse external sources with varying structures and formats.

This newly developed ontology ensured a robust semantic representation of research artefacts, surpassing the limitations of initial alignments with Schema.org and SIOC standards. It was aligned with the main data aggregators, ensuring adherence to state-of-the-art practices in the field. Developed in collaboration with domain experts, the ontology was crafted to achieve several key objectives: formalising the data model with semantic standards, defining controlled vocabularies, establishing connections with external entities, ensuring resource reusability, and maintaining detailed documentation for transparency and extensibility.

The ontology development followed a structured methodology, beginning with a preliminary analysis and employing the Simplified Agile Methodology for Ontology Development (SAMOD). This approach ensured a flexible, iterative development process with comprehensive testing and documentation, guaranteeing the ontology's accuracy and adaptability. Consequently, the ontology enhanced the semantic representation of research artefacts, promoted interoperability, and facilitated collaboration and knowledge reuse across the SSH domain.

Chapter 1: Introduction

This thesis aims to present an ontology that formalises the TRIPLE data model using semantic languages. The TRIPLE project began in October 2019 and involves 22 partners from 15 European countries, coordinated by the French National Center for Scientific Research (CNRS). The project's goal is to develop the GoTriple discovery platform, a multilingual discovery platform for the social sciences and humanities (SSH). GoTriple serves as a central access point for discovering and reusing research artefacts relevant to the wide variety of disciplines within the SSH domain. These research artefacts include publications, research data, project descriptions, and researcher profiles, which are automatically imported from aggregators and source providers, semantically enriched, and linked within GoTriple.

The need to develop this ontology arises from the specific context related to GoTriple's architecture. One of the fundamental architectural components of GoTriple is the Core Pipeline. This layer of the architecture plays a crucial role in gathering data that the platform will make available to end users. The Core Pipeline is responsible for aggregating data from numerous external sources, each adopting its internal data model. These data can vary significantly in structure and format, making their direct use complex without proper standardisation.

To effectively collect and manage this heterogeneous data, it was necessary to define an internal data model for the TRIPLE project. The detailed description of this model is provided in Deliverable D2.5 - Report on Data Enrichment (De Santis, 2022). This document presents the structure of the data model, outlining how the various collected elements are organised and represented internally. Additionally, the Report on Data Enrichment illustrates an initial attempt to align the TRIPLE data model with some classes and properties defined in Schema.org (<https://schema.org/>) and SIOC (<https://www.w3.org/wiki/SIOC>). While these standards are widely used to structure data on the web, the initial alignment, in light of requirements defined by a domain expert, proved insufficient to ensure an effective and comprehensive semantic representation of the TRIPLE model.

To address this limitation, we decided to collaborate with a domain expert to formalise a new ontological model. This new model aims to overcome the shortcomings of the previous one, ensuring an adequate semantic representation of the objects collected by external aggregators. The formalisation of a coherent ontology is essential not only for better internal data representation but also to ensure a high degree of interoperability with widely used ontologies in the semantic web

community and with other similar data aggregators. Specifically, the ontology aims to achieve the following key objectives:

- *Formalization of the Data Model with Semantic Standards:* The ontology must reuse languages accepted among semantic web standards. Additionally, whenever possible, Schema.org, already implemented in the first version of the data model proposed in Deliverable D2.5 - Report on Data Enrichment, should be reused. This ensures consistency with existing standards and facilitates integration with other ontologies.
- *Controlled Vocabularies:* Triple needs to define a set of controlled vocabularies to ensure terminological and semantic consistency among the collected data. These vocabularies will provide a common reference for classifying and describing the data, reducing ambiguities and improving overall information quality.
- *Connection with External Entities:* Some entities must be modelled to connect with external entities. This ability to link internal data with external sources enhances the interoperability and utility of the collected data, allowing for a more comprehensive and integrated view of information.
- *Reusability of Resources:* The Triple ontology must enable the reusability of resources. The entities and properties defined in the ontology should be usable in various contexts and applications, reducing the need to reinvent data structures for new projects and promoting consistency across different implementations.
- *Documentation:* Every decision made during the data model design must be documented and justified within the ontology. This level of transparency is essential to ensure that the choices are understood and shared among various stakeholders and to facilitate future modifications and improvements.
- *Flexibility and Extensibility:* The structure of the Triple ontology must be flexible and extensible. This would allow it to be easily adapted to new knowledge domains or changes in project requirements without compromising the consistency of existing data.
- *Facilitating Collaboration and Knowledge Reuse:* A well-defined ontology promotes collaboration among different teams and organisations, facilitating the reuse and sharing of knowledge and contributing to the creation of a richer and more interconnected data ecosystem. Triple ontology must be understood and used by other researchers or domain experts, ensuring the effective and consistent use of the collected information.

To ensure the development of an ontology that appropriately models the domain and meets the aforementioned conditions, a feasible solution was proposed based on a preliminary analysis, a robust development methodology, and comprehensive supporting documentation. Specifically:

- *Preliminary Analysis:* The analysis of the previously mentioned Deliverable D2.5 - Report on Data Enrichment, along with the formulation of a series of requirements proposed by the domain expert, served as the starting point for the ontology modelling.
- *Simplified Agile Methodology for Ontology Development (SAMOD):* The SAMOD methodology (Peroni 2016) enables the development of a flexible model through an agile and iterative workflow. This workflow is based on the reuse of existing models and patterns and rigorous testing sessions. This approach ensures that the model returns accurate data through queries of varying complexity and manageability.
- *Rich and Detailed Documentation:* As a result of applying SAMOD, the documentation covers every aspect of the model's development, from practical application scenarios to examples of formal queries used to test the model's integrity. This documentation serves as a reference resource for users and any developers interested in implementing further extensions of the model.

The rest of this thesis is structured as follows. In Chapter 2, a review of the literature on topics related to ontology development is presented. This chapter starts with a general definition of data modeling, then addresses data modeling in the context of data aggregators, specifically: Europeana (<https://www.europeana.eu/it>), OpenCitations (<https://opencitations.net/>), ISIDORE (<https://isidore.science/>), OpenAIRE Graph (<https://graph.openaire.eu/>), and OpenAlex (<https://openalex.org/>). Subsequently, data modeling in the context of the Semantic Web is discussed, with a particular focus on ontological modeling. Definitions of ontology and various applications of ontologies modeling research artifacts are provided, including SPAR ontologies (<http://www.sparontologies.net/>), RO-Crate (<https://www.researchobject.org/ro-crate/>), and DCAT (<https://www.w3.org/TR/vocab-dcat-2/>). Additionally, the concept of Linked Data and Tim Berners-Lee's 5-star model are briefly introduced. Chapter 3 describes the process through which the ontology was designed and developed. It begins with a description of the preliminary analysis conducted on Deliverable D2.5 - Report on Data Enrichment, along with the associated results. Next, the ontology design and development process is defined, detailing the methodology used (SAMOD), the reused ontological models, and the additional tools supporting its creation, description, and documentation. Chapter 4 provides a high-level description of the Triple Ontology. Its structure and key primitives are described in detail. It then describes its alignment with the

Functional Requirements for Bibliographic Records (FRBR) model (IFLA Study Group 2008). It then describes in detail how ontologies were reused during the development process. Specifically the following: SKOS (<https://www.w3.org/TR/skos-reference/>), Schema.org (<https://schema.org/>), FOAF (<http://xmlns.com/foaf/spec/>) and PRO (<http://www.sparontologies.net/ontologies/pro>) and some Ontology Design Patterns (ODP) (Gangemi & Presutti 2009). Finally, it describes how the textual data were modelled in the development phase. Exemplary application scenarios are presented to demonstrate potential conceptual issues inherent in the collection and the solutions provided by the proposed model. Chapter 5 details each iteration of the model's development and implementation. The seven iterations address issues defined in collaboration with the domain expert. Chapter 6 concludes the thesis by interpreting the results obtained from the ontological model. It also provides a list of limitations and a series of recommendations for future developments and research. Finally, a brief summary of the thesis is provided.

Chapter 2: State of the Art

The purpose of this chapter is to present the literature related to the key themes for the development of the Triple ontology. In general, ontology development falls within the broader field of data modelling. Therefore, in section 2.1, a definition of data modelling will be provided. However, given the extensive literature and history of this field, it was necessary to narrow the focus to the specific area of our interest: the modelling of Triple, centred on data aggregators.

For this reason, we analysed technical documentation and publications concerning some representative cases of data modelling and data aggregators. The selection of aggregators does not aim to be exhaustive but seeks to examine well-known and widespread cases. These examples will be useful for comparing the results of the Triple development in the final discussion of the outcomes (see Chapter 6.1).

After clarifying the concept of data modelling, we will delve into the technologies implemented for this specific case of modelling, particularly those of the Semantic Web. In Chapter 2.3, an explanation of the Semantic Web will be provided, followed by a description of the fundamental tools for representing knowledge on the web, with particular reference to ontologies (Canali, 2005). Again, the ontologies published on the web are numerous and cover many fields of study (McDaniel, 2019). Therefore, as with data modelling, we have narrowed the analysis to the ontologies relevant to Triple, specifically those that model research artefacts. Section 2.3.3 is dedicated to ontologies that describe entities and relationships related to research artefacts. Here too, the objective is not to be exhaustive but to present notable cases similar to the one addressed in this work.

Finally, in Chapter 2.3.4, Linked Data will be introduced. One of the goals of the Triple development is to formalise connections with entities outside the Triple domain itself.

In summary, the state of the art allows us to examine the key themes that constitute the building blocks of the Triple ontology development: data modelling in data aggregators, ontologies that model research artefacts, and the concept of Linked Data.

2.1 Data Modelling: a definition

The literature on data modelling is extensive, highlighting its significance as a key tool in scientific research (Frigg & Hartmann 2006). Numerous studies have explored the definitions of data modelling and data models, emphasising their functional importance in various contexts and the systems they aim to represent.

In computer science, "data modelling" is traditionally described as "a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints" (Silberschatz et al. 1997, 7). A "data model" is defined as "an abstract, self-contained, logical definition of the data structures, data operators, and other elements that make up the abstract machine with which users interact" (Date 2012, 12). Technically, both terms are predominantly used to describe the process of developing a relational database.

Data modelling in database development involves two main operations: conceptual data modelling and logical data modelling. Conceptual data modelling describes entities, attributes, and relationships in a given domain using an Entity-Relationship (E/R) diagram (Flanders & Jannidis 2015). Logical data modelling then defines the database tables based on the E/R diagram from the previous step (Flanders & Jannidis 2015). The mapping from E/R diagrams to database tables can be done manually or mostly automatically using software, considering the specified entity types, relationship types, and attributes (Teorey et al. 1986).

On the other hand, graph modelling, unlike relational database modelling, leverages the natural structure of graphs to represent and manage data. In graph databases, data entities are modelled as nodes, and relationships between these entities are represented as edges, capturing direct connections in a highly intuitive manner (Pokorný, 2016). This model excels in scenarios where the relationships themselves are as important as the data, such as social networks (Tabassum et al., 2018), recommendation systems (Eirinaki et al., 2018), and biological networks (Girvan & Newman, 2002). Unlike relational databases, which require complex joins to retrieve related data, graph databases can traverse these relationships efficiently, often using graph algorithms that can handle complex queries over large datasets (Wu et al., 2020). Additionally, graph modelling supports dynamic schema evolution, allowing for more flexibility as data structures change over time. This approach contrasts with the rigid schema of relational databases, where changes can be costly and time-consuming (Vera-Olivera et al., 2021). Despite challenges in defining integrity constraints and schema explicitly, graph databases offer a powerful alternative for applications where interconnected data is a primary focus.

2.2 Data Modelling in Data Aggregator

After introducing the concept of data modelling, it is necessary to introduce data modelling within data aggregators. First, we will clarify what is meant by data aggregators, and then illustrate a series of case studies to reference for this work.

The literature on data aggregators is quite limited, if not absent. There is no extended and widely

shared definition of the term within the academic realm. Instead, academic literature focuses more on data ingestion and data integration. Thus, data aggregators are addressed more from the perspective of the tools and methodologies that enable them to function correctly rather than from an architectural viewpoint. For instance, in "Principles of Data Integration" by Doan et al. (Doan et al., 2012), data integration is defined as “a set of techniques that enable building systems geared for flexible sharing and integration of data across multiple autonomous data providers.” From a Big Data Analytics perspective, Hlupić & Puniš (Hlupić & Puniš, 2021) describe data ingestion as the process of collecting and transferring data from various sources into a storage system for future use, often in its raw form without prior transformations. They define data integration as the combination of data from different sources into a unified format through extraction, transformation, and loading (ETL) processes. ETL ensures that the collected data is consistent and ready for comprehensive analysis and reporting.

For the purpose of this thesis, we will provide a definition of data aggregator that closely aligns with the experience of the Triple project and the GoTriple platform. Moreover, the examples analysed in the following sections from a data modelling perspective have been considered in terms of their architectures.

A data aggregator is a system, tool, or service that collects data from multiple sources, cleanses and processes it, and then compiles it into a single dataset for easier analysis and decision-making. Data aggregators are essential in various industries for managing large volumes of data efficiently and deriving actionable insights. The data aggregation process typically involves several steps:

- *Data Collection*: Gathering raw data from various sources such as public records, web scraping, surveys, IoT devices, databases, and data providers.
- *Data Cleansing and Standardization*: Removing duplicates, fixing inconsistencies, and normalising data formats to ensure accuracy and uniformity.
- *Data Enrichment*: Adding contextual information, linking, or reusing external resources.
- *Data Integration*: Combining data from different sources into a unified dataset.
- *Data Storage*: Storing the aggregated data in databases, data warehouses, or data lakes for easy access and analysis.

2.2.1 ISIDORE

ISIDORE is a sophisticated search engine designed for discovering and retrieving publications, digital data, and researcher profiles in the social sciences and humanities (SSH). Launched on December 8, 2010, through a collaboration involving CNRS’s Adonis project, the Center for Direct Scientific Communication, and technology partners Antidot, Mondéca, and Sword, ISIDORE is

maintained by IR* Huma-Num and accessible via <https://isidore.science> (Pouyllau et al, 2021). Its design incorporates several key decisions in data modelling and ontology to enhance the accessibility and usability of academic resources.

According to ISIDORE Documentation (ISIDORE Documentation, 2024), the data model of ISIDORE begins with the extensive harvesting of both textual metadata and full texts from various sources. This collected data undergoes multiple enrichments to create a unified, interconnected dataset. Semantic annotation is a critical process where document metadata is matched with vocabulary entries from multilingual, aligned scientific thesauri. This algorithm-based morphological analysis ensures that resources are linked to relevant scientific concepts, enhancing their discoverability.

Disciplinary categorization further refines this process. ISIDORE employs a semantic classifier, trained on reference corpora, to categorise documents into SSH disciplines using the MORESS vocabulary (ISIDORE Vocabularies, 2024). This classifier benefits from manual categorizations performed by researchers, providing an additional layer of precision and reliability. Author identification is another essential feature; ISIDORE enriches author metadata by cross-referencing with international and national author identifiers (ORCID, VIAF, ISNI, IDHAL, IDRef), ensuring accurate attribution and connectivity.

ISIDORE's indexing system integrates several elements to optimise search efficiency: structured document metadata, open access full texts, semantic annotations, disciplinary classifications, and normalised author data. This comprehensive indexing allows for robust search functionalities, enabling users to filter by discipline, concept, or author.

Documents in ISIDORE are organised into three primary categories, each identified by specific ontological classifications: research documents and data ("primaires"), published documents and data ("secondaires"), and scientific events ("evenementielles"). This structured approach facilitates the seamless organisation and retrieval of diverse academic materials.

The integration with major SSH publication platforms and digital libraries further expands ISIDORE's repository. Platforms like OpenEdition, Cairn, Perseus, Erudit, and digital libraries such as Gallica (BnF) and E-rara contribute to ISIDORE's extensive content, ensuring a rich and diverse collection of resources.

According to ISIDORE Vocabularies specification (ISIDORE Vocabularies, 2024) and ISIDORE Documentation (ISIDORE Documentation, 2024), to standardise and categorise the vast array of documents, ISIDORE employs a detailed ontology of document types, aligned with international standards such as COAR, BIBO, RDFS, DCAT, and Wikidata. This alignment, particularly with the NAKALA repository's types within the Huma-Num infrastructure, supports interoperability and

enhances the aggregation process. The ISIDORE ontology is available online in XML (SKOS/RDF), and its labels are accessible in English, French, and Spanish.

2.2.2 Europeana

Europeana is a digital platform that provides access to millions of digitised items from Europe's cultural and scientific heritage, including books, music, artworks, and videos from across Europe. To manage, integrate, and publish the diverse metadata from various cultural heritage institutions such as museums, libraries, archives, and audiovisual collections, Europeana developed the Europeana Data Model (EDM). This model supersedes the Europeana Semantic Elements (ESE) and offers a more flexible and expressive framework for metadata representation.

According to the definition of the Europeana Data Model (Europeana, 2016), the EDM (Europeana Data Model) was developed to address several key challenges and requirements inherent in managing and publishing metadata from diverse cultural heritage institutions. The primary objective was to transcend the limitations of the previous Europeana Semantic Elements (ESE) model by providing a more flexible and expressive framework. This flexibility allows EDM to accommodate various metadata standards used by museums, libraries, archives, and audiovisual collections, ensuring that the richness and specificity of each provider's data are preserved.

In developing EDM, several critical design choices were made. Firstly, EDM was designed to support the full richness of metadata from content providers while enabling data enrichment from external sources. This was achieved through the use of RDF (Resource Description Framework) and other Semantic Web technologies, allowing for a flexible, open, cross-domain framework. This framework can integrate with community-specific standards like LIDO for museums, EAD for archives, and METS for digital libraries.

Secondly, EDM introduces a clear distinction between the "provided cultural heritage object", its digital representations, and the aggregations of these elements. This separation ensures that the original objects and their digital manifestations can be managed and represented distinctly, catering to both users' interests and system requirements.

To handle multiple views on the same object from different providers, EDM employs proxies. Proxies allow Europeana to maintain different, potentially conflicting descriptions of an object while keeping track of each description's provenance. This mechanism is crucial for integrating metadata from various sources without compromising the original data's integrity.

EDM also incorporates mechanisms for rich, contextual metadata. It supports both object-centric and event-centric descriptions, allowing metadata to focus on the object's various attributes or the events in its lifecycle. Contextual entities such as agents, places, time spans, and concepts can be linked to the objects, providing deeper contextualization and enrichment.

Moreover, EDM adheres to several modelling principles of the Semantic Web, such as the reusability and linkage of data. It does not impose a fixed schema but serves as an anchor to which finer-grained models can be attached. This approach facilitates interoperability at the semantic level while retaining the original expressivity and richness of the data. Providers are encouraged to use publicly accessible vocabularies and to submit both detailed and generalised metadata to ensure broad compatibility and enrich the user experience.

Finally, EDM's design supports complex objects and hierarchical relationships, accommodating the needs of digital representations that consist of multiple parts, like books with chapters or archival records. This hierarchical structuring is crucial for detailed and accurate metadata representation.

In summary, the development of EDM was guided by the need for flexibility, richness, and interoperability in metadata representation. By leveraging Semantic Web technologies, supporting multiple views through proxies, enabling rich contextual descriptions, and accommodating complex object structures, EDM ensures that Europeana can effectively manage and enrich a diverse array of cultural heritage metadata.

2.2.3 OpenAlex

OpenAlex was created as an open-source alternative to the Microsoft Academic Graph (MAG), aiming to provide a comprehensive and transparent index of scholarly works, authors, venues, institutions, and concepts (Priem, J et al, 2022). According to OpenAlex Technical Documentation (OpenAlex Technical Documentation, 2024), the data model of OpenAlex is designed as a heterogeneous directed graph with five primary entity types, each playing a distinct role in representing scholarly data .

- *Works*: These are scholarly documents such as journal articles, books, datasets, and theses. OpenAlex indexes approximately 209 million works, with around 50,000 added daily. The Canonical External ID (CEID) for works is the Digital Object Identifier (DOI), used for about half of the indexed works. Works are particularly important because they form the core around which other entities (authors, venues, institutions, and concepts) are connected. Metadata is parsed from structured sources like Crossref and PubMed, as well as unstructured sources from publisher landing pages .
- *Authors*: Defined as individuals who create scholarly works, OpenAlex indexes around 213 million authors, adding thousands daily. The CEID for authors is the Open Researcher and Contributor ID (ORCID). Authors are connected to works via the "authorship" object, which formalises the affiliation of authors with institutions and their contributions to scholarly

works. ORCID is used to help algorithmically disambiguate authors when available, alongside publication records and citation histories.

- *Venues*: These are the places where works are published, including journals, conferences, and repositories. OpenAlex indexes about 124,000 venues, with the CEID being the Linking ISSN (ISSN-L), which groups all ISSNs associated with a publication. Works may be hosted in multiple venues and versions, such as preprints and versions of record. A fingerprinting algorithm is employed to match these versions and identify the primary host, while also determining the version and license of each copy where possible.
- *Institutions*: Organisations to which authors are affiliated, OpenAlex indexes about 109,000 institutions. The CEID for institutions is the Research Organization Registry ID (ROR ID). Institutions are linked to works through affiliation data parsed from structured and unstructured sources. This data is normalised using a two-step algorithm combining rules-based and machine-learning stages.
- *Concepts*: Abstract ideas that works are about, OpenAlex indexes around 65,000 concepts. The CEID for concepts is the Wikidata ID, ensuring unique identification and a hierarchical structure. Concepts are assigned to works using an automated classifier trained on MAG's corpus, with approximately 85% of works having at least one concept assigned.

Despite its comprehensive coverage, OpenAlex is still improving its parsing, normalisation, and disambiguation processes, particularly for authors and institutions. Future enhancements will focus on including metadata about funding sources and corresponding authors, and ensuring data quality and reliability compared to other scholarly knowledge graphs (OpenAlex Technical Documentation, 2024).

2.2.4 OpenCitations

OpenCitations is an innovative infrastructure organisation dedicated to the publication of open bibliographic and citation data using Semantic Web technologies. Established to provide an open alternative to proprietary citation indexes such as Web of Science and Scopus, OpenCitations offers free access to scholarly citation data. This infrastructure is crucial for promoting transparency and reproducibility in bibliometric analysis by making the source data openly available (Peroni & Shotton, 2019). According to Massari et al. (Massari et al., 2024) The organisation maintains several key resources:

- *OpenCitations Indexes*: Including several indexes such as the OpenCitations Index of Crossref Open DOI-to-DOI Citations (COCI), PubMed Open Citation Index (POCI), DataCite Open DOI-to-DOI Citation Index (DOCI), and the Crowdsourced Open Citations Index (CROCI).
- *OpenCitations Meta*: A database for open bibliographic metadata of scholarly publications involved in the citations indexed by the OpenCitations infrastructure. It assigns globally persistent identifiers known as OpenCitations Meta Identifiers (OMIDs) to all bibliographic resources, enhancing performance by eliminating reliance on API calls to external resources. This system includes automated data curation processes for error correction, metadata enrichment, and provenance tracking, ensuring data integrity and transparency.

OpenCitations employs Semantic Web technologies to publish citation data in a structured, machine-readable format, enhancing the interoperability and usability of the data across different platforms and applications.

The core of OpenCitations' data infrastructure is the OpenCitations Data Model (OCDM), which meticulously models bibliographic and citation information. The OCDM encompasses various classes of entities and their relationships, effectively capturing the intricate details of bibliographic references.

According to Daquino et al. (Daquino et al., 2023), the key elements of the OCDM are as follows:

1. Published Bibliographic Resources

- Represents the actual intellectual content of published works, such as journal articles, books, and conference papers.
- This class captures the essential details of a work, including title, publication date, and authorship.

2. Manifestations

- Defines the specific physical or digital formats in which a bibliographic resource is available.
- Examples include print versions, PDF files, and ePub formats.

3. Bibliographic References

- Represents the references listed within a citing entity that point to another bibliographic resource.
- This class includes attributes such as the reference list entry, in-text citation pointers, and the context of the reference.

4. Responsible Agents

- Identifies individuals or organisations associated with the bibliographic resources.
- Examples include authors, editors, publishers, and institutions.
- This class captures roles and affiliations, providing a comprehensive view of the responsible entities involved in the creation and dissemination of scholarly works.

5. Roles

- Describes specific roles held by agents with respect to bibliographic resources.
- Examples include authorship, editorship, and funding roles.
- This class allows for the temporal aspect of roles, indicating the period during which an agent held a particular role.

6. Citations

- Represents the directional relationships between citing and cited resources.
- This class includes properties to capture the citation's context, purpose, and type (e.g., self-citation, co-citation).
- Citations are treated as first-class entities, allowing for detailed bibliometric analysis and visualisation of citation networks.

7. External Identifiers

- Captures unique identifiers associated with bibliographic entities, such as DOIs, ORCIDs, PubMed IDs, and OCIs.
- This class ensures the precise identification and disambiguation of entities across different datasets and platforms.

8. In-Text Reference Pointers

- Represents the textual devices (e.g., “[1]” or “Peroni & Shotton 2019”) embedded in the text of a document that point to bibliographic references.
- This class captures the precise location and context of in-text references within the citing work.

9. Annotations

- Used to link in-text reference pointers to their corresponding bibliographic references and citations.
- This class provides a detailed description of the function and context of each reference, supporting nuanced citation analysis.

10. Discourse Element

- Defines the context in which in-text reference pointers occur, encompassing various units of text such as sentences, paragraphs, or sections.

The use of RDF (Resource Description Framework) and ontologies from the SPAR (Semantic Publishing and Referencing) family ensures that the data is not only structured and interoperable but also reusable and accessible. This adherence to the FAIR principles (Findable, Accessible, Interoperable, and Reusable) enhances the usability of the data for various stakeholders, including researchers, librarians, funders, and academic administrators.

The structured representation of citations as first-class data entities in the OCDM allows for advanced bibliometric analyses. It supports the creation of citation networks, tracking of citation provenance, and the integration of citation data into other systems and applications. This approach significantly improves the accuracy and depth of bibliometric research, offering new insights into scholarly communication patterns.

This comprehensive approach not only supports advanced bibliometric analysis, but also democratises access to citation data, fostering inclusivity in scholarly research. By adhering to FAIR principles and employing Semantic Web technologies, OpenCitations ensures that citation data is findable, accessible, interoperable, and reusable, thereby promoting open scholarship and facilitating new insights into academic communication patterns.

For instance, Butt et al. (Butt et al., 2021) developed a workflow using publicly available citation metadata from Crossref and OpenCitations to create scientific networks, highlighting techniques like centrality analysis and community detection. Similarly, Gianmarco et al. (Gianmarco et al., 2022) conducted an experiment to measure the coverage of Digital Humanities (DH) publications in various bibliographic data sources, including OpenCitations. Their study revealed strong connections between DH and fields such as Computer Science, Linguistics, and Psychology, underscoring the interdisciplinary nature of DH research.

2.2.5 OpenAIRE

The OpenAIRE initiative is dedicated to fostering an Open Science e-Infrastructure in Europe, promoting open access to research outcomes. This initiative operates an infrastructure that supports the sharing, re-use, and management of research outputs, enhancing the dissemination and impact of scientific knowledge. Central to this infrastructure is the OpenAIRE Graph, a comprehensive graph-like structure where research products are semantically linked. The development of the OpenAIRE data model and ontology was guided by the need to support diverse types of research outputs and their relationships within the research lifecycle.

According to documentation of Manghi et al. (Manghi et al., 2019), the OpenAIRE data model is heavily inspired by established standards such as DataCite and CERIF, ensuring compatibility and interoperability within the broader research ecosystem. The primary entities within this model include Results, Persons, Organizations, Funders, Funding Streams, Projects, and Data Sources.

- *Results* represent the outcomes of research activities, including datasets and publications. Each result can manifest in multiple instances, which may have different access rights (open, embargoed, restricted, or closed) and provenance information, linking back to their respective data sources. This design choice ensures that multiple manifestations of the same research output are accurately represented and accessible.
- *Persons* are individuals involved in the research process, such as authors and project coordinators. This entity captures the various roles individuals play, ensuring that contributions are properly attributed and searchable.
- *Organisations* encompass companies, research centres, and institutions that participate in research projects or manage data sources. This broad definition allows the model to represent the diverse types of organisations involved in research and their roles within projects and data management.
- *Funders* and *Funding Streams* represent the entities that provide research funding and the specific funding schemes they manage. By capturing these relationships, the model can trace the financial support behind research outputs, aiding in the assessment of research impact and return on investment.
- *Projects* are defined as research endeavours funded by specific funding streams. This entity links projects to their outcomes (Results), funders, and the individuals and organisations involved, providing a comprehensive view of the research lifecycle.
- *Data Sources* are the repositories and journals from which OpenAIRE collects data to populate the ISG. Each object in the ISG is associated with its data source, ensuring provenance and traceability. This association is critical for maintaining the integrity and credibility of the aggregated information.

2.3 Data Modelling in the Semantic Web

2.3.1 The Semantic Web

The Semantic Web, a concept initially introduced by Tim Berners-Lee (Berners-Lee et al., 2001), aims to structure and organise the unstructured information available on the Web (Zöllner-Weber, 2009). While the World Wide Web provides accessible information, it lacks comprehensibility for computer programs and software systems, transforming mere information into meaningful knowledge. For instance, search engines can identify keywords but fail to understand their semantics, often delivering irrelevant results alongside accurate ones due to their inability to grasp meaning (Berners-Lee, 1998). Artificial agents also struggle due to this semantic gap, as they cannot effectively locate and utilise web-based services without a shared understanding of the services' functions and usage (Berners-Lee et al., 2001).

The Semantic Web seeks to address these issues by enabling software agents to achieve a common semantic ground through shared metadata organised in controlled vocabularies and ontologies. This framework facilitates a World Wide Web infrastructure based on semantic data integration, supporting software applications in efficiently searching for and retrieving information across multiple domains. Consequently, this allows for more structured and sophisticated tasks by human users and other agents (Doerr & LeBoeuf, 2006).

Semantic Web technologies offer solutions to challenges in data integration, knowledge formalisation, information retrieval, and mapping (Aljalbout & Felquet, 2018). A "Web of data" enables humanities researchers to leverage these technologies for retrieving relevant answers to diverse research questions, ensuring interoperability, usefulness, openness, dissemination, communication, sharing, and integration of their projects' data and metadata on the Web (Antezana et al., 2009; Meroño-Peñuela, 2013).

2.3.2 Ontology: a definition

The vision of the Semantic Web relies on data models known as "ontologies." In their paper, Guarino et al. (Guarino et al., 2009) trace the history and usage of this term in both philosophy and computer science. Initially, ontology, understood as a data structure, is defined by Gruber (Gruber, 1993) as an "explicit specification of a conceptualization." A few years later, Uschold & Gruninger (Uschold & Gruninger, 1996) describe it as "a formal vocabulary, a semantic model of shared knowledge, which comprises a set of concepts, their definitions, and semantic interrelationships."

Borst (1997) builds on Gruber's definition, stating that an ontology is a "formal specification of a shared conceptualization." These definitions are integrated by Studer et al. (1998).

Guarino and colleagues further elaborate on the terms constituting a comprehensive definition of ontology. Specifically, an ontology is an explicit and partial specification of a shared conceptualization formalised in a logical theory, where:

- *Conceptualization* refers to an abstract model of a portion of reality or domain, comprising concepts and relationships governed by formal rules.
- *Formal* indicates the model's machine-readability.
- *Partial* denotes that an ontology represents a necessarily limited perspective on a portion of reality or domain.
- *Explicit* pertains to the defined concepts, relationships, and formal rules constituting the ontology.
- *Shared* signifies that the knowledge captured by an ontology is accepted by multiple people, communities, etc.
- *Specification* means the ontology is expressed through a logical language, reducing semantic ambiguity.
- *Logical theory* includes a vocabulary (a set of human-readable definitions of concepts and relationships describing the reality or domain) and a set of logical rules (axioms), both integrated into a taxonomy.

Ontologies can be specified through standards developed by the World Wide Web Consortium, such as the Web Ontology Language (OWL 2) (Hitzler et al. 2009) and, to some extent, the Resource Description Framework Schema (RDFS) (Brinkley and Guha 2004). These standards provide data models where data is organised and visualised in a graph-based structure. This graph consists of statements about the domain of knowledge structured in triples in the form of "subject-predicate-object." In the graph, the "subject" and "object" are nodes connected by a directed arc labelled as "predicate." The subject denotes a resource, the object denotes another resource or an attribute of the subject, and the predicate expresses a semantic relationship between the two. Each subject, predicate, and object is identified by its Uniform Resource Identifier (URI) (Berners-Lee et al. 2005), ensuring unambiguous and persistent resource identification.

For example, consider the following triples, each representing a statement expressed in pseudocode:

- <John> <is a> <Person> ("John is a person").
- <John> <knows> <Peter> ("John knows Peter").

- `<Peter> <has age> <21>` ("Peter is 21 years old").

In the first and second triples, `<John>` is the subject, `<is a>` and `<knows>` are predicates, and `<Person>` and `<Peter>` are objects. In the third triple, `<Peter>` is the subject, `<has age>` is the predicate, and `<21>` is the object. Additionally, each subject and predicate has a URI:

- `<John>`: `<https://example.org/people/john>`
- `<Peter>`: `<https://example.org/people/peter>`
- `<is a>`: `<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>`
- `<knows>`: `<http://xmlns.com/foaf/0.1/knows>`
- `<has age>`: `<http://xmlns.com/foaf/0.1/age>`

Moreover, `<Person>` has a URI: `<http://xmlns.com/foaf/0.1/Person>`. Thus, the triples can be expressed as:

1. `<https://example.org/people/john>`
`<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>`
`<http://xmlns.com/foaf/0.1/Person>`
2. `<https://example.org/people/john>`
`<http://xmlns.com/foaf/0.1/knows>`
`<https://example.org/people/peter>`
3. `<https://example.org/people/peter>`
`<http://xmlns.com/foaf/0.1/age>` `<"21">`

The URIs `<https://example.org/people/john>` and `<https://example.org/people/peter>` are illustrative, while the others belong to existing vocabularies: `<http://xmlns.com/foaf/0.1/Person>`, `<http://xmlns.com/foaf/0.1/knows>`, and `<http://xmlns.com/foaf/0.1/age>` from the Friend Of A Friend vocabulary (FOAF), and `<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>` from the RDF Concepts Vocabulary. A resource can be the subject of multiple triples, the object of multiple triples, and so forth.

Formal rules and constraints can be applied to concepts and relationships to infer new information not explicitly encoded. For example, the predicate `<http://xmlns.com/foaf/0.1/knows>`, as defined in its vocabulary, implies that any

value it points to is a person. Thus, the statement `<Peter> <is a> <Person>` can be inferred even if not explicitly encoded.

Ontologies offer significant benefits. Auer & Herre (Auer & Herre, 2007) argue that ontologies capture the semantics of knowledge in a format designed for easy maintenance and efficient processing by reasoning algorithms. The organisation of knowledge and the knowledge itself about modelled objects are expressed clearly and meaningfully, making the information inferable, reusable, and accessible to scientific communities, researchers, companies, and the general public. Using an ontology facilitates new scholarly inquiries, such as searching implicit information based on automatic reasoning or linking a resource to multiple other web resources. Overall, ontological data modelling organises discrete facts into a coherent information system where semantic information is structured, managed, and made available to a wider audience.

2.3.3 Ontologies for modelling research artefacts

2.3.3.1 Semantic Publishing and Referencing (SPAR) ontologies

According to Peroni & Shotton (Peroni & Shotton, 2018) SPAR (Semantic Publishing and Referencing) Ontologies represent a suite of interrelated and complementary OWL 2 DL ontologies specifically designed to model various aspects of the scholarly publishing domain. The primary objective of these ontologies is to enhance the discoverability, accessibility, and usability of scholarly data through machine-readable formats. This section provides an overview of SPAR ontologies, their components, and their relevance in modelling research artefacts.

SPAR Ontologies were conceived to address the limitations of existing bibliographic models and to support the full spectrum of the scholarly publishing lifecycle. They aim to provide a robust and extensible framework that accommodates the intricate nature of scholarly communication, from metadata description and citation analysis to the detailed modelling of publication workflows and roles.

SPAR Ontologies consist of various modules, each targeting specific elements of the scholarly publishing process:

- FaBiO (FRBR-aligned Bibliographic Ontology): Models bibliographic entities and their relationships.
- FRBR-DL (Essential FRBR in OWL2 DL Ontology): Offers an OWL representation of FRBR concepts.

- DoCO (Document Components Ontology): Provides vocabulary for document structures and elements.
- DEO (Discourse Elements Ontology): Describes rhetorical structures within documents.
- DataCite Ontology: Manages identifiers for bibliographic resources and related entities.
- CiTO (Citation Typing Ontology): Captures citation intent and context.
- BiRO (Bibliographic Reference Ontology): Details references and reference lists.
- C4O (Citation Counting and Context Characterisation Ontology): Characterises in-text citations and contexts.
- PRO (Publishing Roles Ontology): Describes roles of agents in the publication process.
- PSO (Publishing Status Ontology): Tracks the status of documents through the publication lifecycle.
- PWO (Publishing Workflow Ontology): Models steps in publishing workflows.
- SCoRO (Scholarly Contributions and Roles Ontology): Defines contributions and roles in academic activities.
- FRAPO (Funding, Research Administration and Projects Ontology): Manages academic administrative data.
- BiDO (Bibliometric Data Ontology): Encodes bibliometric data and evaluations.
- FiveStars (Five Stars of Online Journal Articles Ontology): Rates articles using a five-star system.

SPAR Ontologies offer several benefits for modelling research artefacts:

- Expressiveness and Specificity: SPAR ontologies provide detailed, semantically rich descriptions of research artefacts, capturing nuances that general bibliographic ontologies often miss.
- Interoperability: By adhering to established standards and incorporating existing vocabularies like DCTerms, SKOS, and FOAF, SPAR ontologies ensure interoperability across diverse systems and datasets.

- **Enhanced Discoverability:** Semantic annotations using SPAR ontologies improve the discoverability of research artefacts by enabling more sophisticated queries and reasoning over the data.
- **Modularity and Reusability:** The modular nature of SPAR ontologies allows for selective adoption and integration, ensuring that specific needs can be met without unnecessary complexity.
- **Support for Scholarly Communication:** SPAR ontologies facilitate the detailed modelling of the scholarly communication process, including citation contexts, publication workflows, and author contributions.

SPAR ontologies can be used to model cases such as the following:

- **Metadata Description:** Using FaBiO and DoCO, researchers can provide detailed descriptions of articles, book chapters, and other scholarly works, specifying components like abstracts, sections, and figures.
- **Citation Analysis:** CiTO and C4O enable the precise characterization of citations, capturing citation intent, context, and in-text references, which is invaluable for citation network analysis.
- **Publishing Workflows:** PWO and PSO support the modelling of publishing workflows, tracking the status of manuscripts from submission through peer review to publication, thus ensuring transparency and traceability.

SPAR Ontologies significantly enhance the modelling and interoperability of scholarly data. They provide a comprehensive, modular framework that addresses various aspects of scholarly publishing, from metadata and citation analysis to workflow and role management. Their adoption and use can lead to more efficient, discoverable, and reusable scholarly communications, ultimately contributing to scientific research and collaboration.

2.3.3.2 Research Object Create (RO-Crate)

According to Metadata Specification of Sefton et al. (Sefton et al., 2023) the RO-Crate data model is grounded in the principles of Linked Data (see chapter 2.3.4), making extensive use of ontologies to represent and interlink research data and metadata. The ontologies used in RO-Crate provide a structured, machine-readable way to describe various entities and their relationships within a research dataset. Here are the core characteristics of the ontology used in RO-Crate:

RO-Crate employs JSON-LD (JSON for Linked Data) as its primary format for representing metadata. JSON-LD allows embedding linked data in JSON, making it both human-readable and machine-processable. This format is essential for enabling interoperability and linking data across different systems and platforms.

Schema.org is the foundational vocabulary for RO-Crate. It is widely adopted on the web and supported by major search engines, which enhances the discoverability of RO-Crate metadata. Schema.org provides a broad set of types and properties that RO-Crate leverages to describe datasets, files, and related entities like people, organisations, and software.

While Schema.org forms the basis, RO-Crate also adapts and extends other ontologies and vocabularies as needed. For instance, terms from the Portland Common Data Model (PCDM) are used to describe collections and digital objects. This approach ensures that RO-Crate can describe a wide range of research outputs comprehensively.

RO-Crate distinguishes between different types of entities within a dataset:

- *Root Data Entity*: Represents the entire dataset or research object.
- *Data Entities*: Individual files or directories included in the dataset.
- *Contextual Entities*: Provide additional context and provenance, such as information about people, organisations, places, and equipment involved in the creation and use of the dataset.

Each of these entities is described using appropriate types and properties from the chosen vocabularies. For example, a file might be described as a `MediaObject`, while a person involved in the research might be described using the `Person` type from Schema.org.

RO-Crate supports the extension of its metadata model with custom terms when existing vocabularies are insufficient. Implementers can define new terms within their own namespaces or use the RO-Crate public namespace to ensure stability and avoid conflicts. This flexibility allows RO-Crate to adapt to specific needs of different research domains while maintaining interoperability with other datasets and systems.

RO-Crate emphasises the use of persistent, globally unique identifiers (URIs) for all entities. This practice aligns with Linked Data principles and ensures that each entity can be reliably referenced and interconnected. Recommended identifier schemes include DOIs for datasets, ORCID IDs for researchers, and ROR IDs for organisations.

To facilitate ease of use and processing, RO-Crate JSON-LD metadata is presented in a flattened and compacted form. This means that nested entities are represented as separate entries in a single graph array, simplifying the structure and making it more accessible for both humans and machines.

The ontology characteristics of the RO-Crate data model are designed to provide a robust, flexible, and interoperable framework for describing research data. By leveraging established vocabularies like Schema.org, supporting extensions, and adhering to Linked Data principles, RO-Crate ensures that research objects are well-described, discoverable, and reusable across different contexts and platforms.

2.3.3.3 Data Catalog Vocabulary (DCAT)

According to the technical documentation of Albertoni et al. (Albertoni et al., 2020) DCAT (Data Catalog Vocabulary) is an RDF vocabulary designed for representing data catalogues. The vocabulary is structured around six primary classes: Catalog, Resource, Dataset, Distribution, DataService, and CatalogRecord.

- *Catalogue* represents a catalogue that contains metadata records describing datasets or data services. This entity captures the collection of metadata about datasets or data services, facilitating their discovery and reuse.
- *Resource* is a generalised class for any resource described in a catalogue, serving as the parent class for more specific entities like Dataset and DataService. This approach provides an extension point for defining a catalogue of any resource type, enabling flexibility and extensibility.
- *Dataset* represents a collection of data curated or published by a single agent. Datasets can include various data forms such as numbers, text, images, and multimedia. Each Dataset can be associated with one or more
- *Distributions*, which are accessible forms of the dataset, such as downloadable files. This separation between the conceptual entity (Dataset) and its physical representations (Distributions) allows for clear and organised metadata descriptions.
- *DataService* represents a collection of operations accessible through an API, providing access to datasets or data processing functions. This entity supports various operations, including data selection, extraction, combination, and transformation. It also allows for describing data services that provide dynamic data or real-time data processing capabilities.

- *CatalogRecord* represents metadata about catalogue entries, capturing information such as who added the item and when. This entity is used to record provenance information about entries in a catalogue, making it possible to track the history and origin of the catalogued items.

DCAT is formalised as an OWL2 ontology using RDF-Schema, with each class and property denoted by an IRI. The vocabulary recommends using global identifiers to enhance collaborative annotation and linking, and discourages the use of blank nodes due to their limitations in the linked data context. Examples provided in the vocabulary use Turtle syntax, demonstrating specific capabilities of DCAT without showing all potential properties and links.

In developing the DCAT model and ontology, several key choices were made:

1. **Modular Structure:** By defining specific classes for different types of resources (Catalog, Resource, Dataset, Distribution, DataService, CatalogRecord), DCAT achieves a clear and organised structure that facilitates the representation and management of data catalogues.
2. **Separation of Conceptual and Physical Entities:** Distinguishing between conceptual entities (like Dataset) and their physical representations (like Distribution) allows for detailed and flexible metadata descriptions.
3. **Extensibility:** The use of a generalised Resource class and the ability to define additional subclasses in DCAT profiles or applications provides extensibility, enabling the inclusion of various resource types beyond the default scope.
4. **Standardisation and Interoperability:** Formalizing DCAT as an OWL2 ontology with RDF-Schema ensures compliance with established standards, promoting interoperability across different systems and domains.
5. **Provenance Tracking:** The inclusion of CatalogRecord for metadata about catalogue entries supports provenance tracking, ensuring transparency and traceability of data within catalogues.

Overall, DCAT's design choices reflect a focus on modularity, clarity, extensibility, and standardisation, ensuring it serves as a robust framework for data catalogue representation and management.

2.3.4 Linked Data

Linked Data refers to a set of best practices for publishing and interconnecting structured data on the Web, enabling unified linking and querying of data from various sources. This concept, introduced by Tim Berners-Lee, the inventor of the World Wide Web, is based on standards for encoding, exchanging, and reusing structured metadata. These standards, such as the Resource Description Framework (RDF), enable semantic interoperability between applications sharing information on the Web.

The basic principles of Linked Data were first defined in an article by Tim Berners-Lee (Berners-Lee, 2009), then reiterated during a TED conference in 2009 (Berners-Lee, 2011). He outlined the following points:

1. *Use URIs to Identify Things*: Every entity, such as a person, place, or concept, should have a unique identifier in the form of a URI.
2. *Use HTTP URIs*: These URIs should be dereferenceable, meaning they can be looked up on the Web using standard HTTP protocols.
3. *Provide Useful Information*: When a URI is dereferenced, it should provide useful information about the entity, formatted in standard data models like RDF.
4. *Include Links to Other URIs*: This creates a web of data by linking related entities across different datasets.

Considering these principles, several components are seen as constituents of Linked Data (Wood, David et al, 2014):

- *URI*: Uniform Resource Identifiers uniquely identify entities on the web.
- *HTTP*: The protocol used to dereference URIs and access information.
- *Structured Data Using Controlled Vocabulary Terms*: Data expressed in RDF serialisation formats such as RDFa, RDF/XML, N3, Turtle, or JSON-LD.
- *Linked Data Platform*: A specification defining integration patterns for building RESTful HTTP services capable of reading/writing RDF data.

To illustrate Linked Data principles, consider a simple example involving a single book and its author. The book, "The Old Man and The Sea," written by Ernest Hemingway, can be represented using RDF triples in Turtle syntax (see listing 1). The book is identified with the URI `ex:book1`

and defined as an instance of `schema:Book`. This resource has a title "The Old Man and The Sea" and a creator linked to the URI `ex:ernest_hemingway`. The author, Ernest Hemingway, is represented by `ex:ernest_hemingway`, defined as a `schema:Person`. This resource includes properties such as the author's name "Ernest Hemingway," birth date "1899-07-21," and birth place linked to the DBpedia resource for Oak Park, Illinois (`dbpedia:Oak_Park,_Illinois`). Additionally, the author resource is linked to the DBpedia resource for Ernest Hemingway (`dbpedia:Ernest_Hemingway`). This example demonstrates how unique URIs identify entities, HTTP URIs make the data dereferenceable, useful information is provided about each entity, and links to external datasets enrich the data and connect it to a broader web of information.

```
@prefix schema: <http://schema.org/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix ex: <http://example.org/> .
@prefix dbpedia: <http://dbpedia.org/resource/> .

# Book Information
ex:book1 a schema:Book ;
    dcterms:title "The Old Man and The Sea" ;
    dcterms:creator ex:ernest_hemingway .

# Author Information
ex:ernest_hemingway a schema:Person ;
    schema:name "Ernest Hemingway" ;
    schema:birthDate "1899-07-21"^^xsd:date ;
    schema:birthPlace dbpedia:Oak_Park,_Illinois ;
    schema:sameAs dbpedia:Ernest_Hemingway .
```

Listing 1: Example of a resource described in RDF linked to a resource described on dbpedia.

Practical applications of Linked Data can be seen in several prominent cases. DBpedia (<https://www.dbpedia.org/>), for instance, extracts structured information from Wikipedia and makes it available on the Web. Each Wikipedia article, such as the one about Berlin, is transformed into a structured RDF dataset, linking various attributes like population, geographic coordinates, and related entities. Similarly, the BBC (<https://www.bbc.co.uk/ontologies>) employs Linked Data to manage and deliver content for its websites, such as BBC Programmes and BBC Music. By linking data from diverse sources like MusicBrainz (<https://musicbrainz.org/>) and DBpedia, the BBC efficiently generates web pages for thousands of programs and artists, ensuring up-to-date and enriched content. Additionally, governments, such as those of the United States and the United Kingdom, have embraced Linked Data to publish open government data. For instance, data.gov.uk

(<https://www.data.gov.uk/>) offers datasets on various domains, including transportation and healthcare, in linked data formats, facilitating transparency and public access.

Building on the concept of Linked Data, Linked Open Data (LOD) introduces the notion of openness. LOD refers to Linked Data that is published under an open license, permitting anyone to freely access, use, modify, and share the data. The key emphasis of LOD is not only on the technical aspects of interoperability and linkability but also on ensuring that the data is accessible and available to everyone without restrictions (Wood, David et al, 2014). To evaluate the quality of openness of Linked Data, Tim Berners-Lee proposed a star rating system (Berners-Lee, 2009):

- ★ *Data is available on the Web, in whatever format (e.g., a scanned image):* This is the most basic level. The data is made publicly available on the web, in any form. Even if it's a scanned image of a document, making it available online is the first step. While this makes the data accessible, it is not machine-readable, which limits its usability for data integration and automated processing.
- ★★ *Data is available as machine-readable structured data (e.g., an Excel spreadsheet):* At this level, data is provided in a format that can be easily processed by a computer. Common formats include CSV, Excel, or XML. This is an improvement over raw data because it allows software to parse and manipulate the data programmatically. However, the data might still be in a proprietary format, limiting its interoperability.
- ★★★ *Data is available in a non-proprietary format (e.g., CSV instead of Excel):* Non-proprietary formats ensure that data can be used without the need for specific software. CSV is a common choice because it is simple and widely supported. This step enhances the data's accessibility and longevity, as it can be read and used by any tool or application that supports open standards.
- ★★★★ *Data is published using open data standards from the World Wide Web Consortium (W3C):* This involves using W3C standards such as RDF and SPARQL. RDF (Resource Description Framework) is a framework for representing information about resources on the web, and SPARQL is a query language for databases. By adhering to these standards, data becomes part of a larger ecosystem, enabling it to be linked with other datasets and queried in a uniform manner. This greatly enhances the potential for data reuse and integration.
- ★★★★★ *All of the above, plus links to other people's data to provide context:* This is the highest level of linked data quality. It not only meets all the criteria mentioned above but

also includes links to other datasets. These links create a web of interconnected data, where users can navigate from one dataset to related datasets seamlessly. This network of linked data is what makes it truly powerful, as it allows for richer and more comprehensive data exploration and analysis.

The adoption of Linked Data brings various benefits, including enhanced data integration, improved data discoverability, machine readability, and interoperability. By using standardised formats and protocols, data from various sources can be combined and used in diverse applications without compatibility issues. Additionally, the open and connected nature of Linked Data fosters innovation by providing a rich, interoperable data environment. Developers and researchers can build new applications and services on top of existing data, driving advancements in fields such as smart cities, healthcare, and e-commerce. In summary, Linked Data enhances the discoverability and usability of data by connecting related information across different sources, ultimately facilitating a more interconnected and accessible web of information.

Chapter 3: Methodology

This chapter summarises the tools and methods used for the development of the ontology. Section 3.1 aims to provide a more detailed overview of the project within which this work is situated: TRIPLE. In particular, it analyses the GoTriple platform, the main outcome of the project. After some general introductory information about the platform, its architecture is illustrated. In Section 3.2, we will move on to the document that served as the starting point for the requirements analysis for ontology development: the D2.5 Report On Data Enrichment (De Santis, 2022). Specifically, we will outline the useful information extracted from this document, which essentially provides the initial version of the ontology.

3.1 Preliminary Analysis

3.1.1 The case of GoTriple

GoTriple is an innovative multilingual discovery platform dedicated to the social sciences and humanities (SSH). The platform offers centralised access for discovering and reusing research artefacts pertinent to a wide variety of disciplines within the SSH. These artefacts include publications, research data, project descriptions, and researcher profiles, which are automatically imported from aggregators and data providers, semantically enriched, and integrated within the GoTriple platform.

The platform allows users to:

- *Discover and Reuse Academic Resources*: SSH open academic resources, currently dispersed across various local and discipline-specific repositories, can be discovered and reused in multiple European languages.
- *Connect with Researchers and Projects*: Users can find and connect with other researchers and projects, overcoming disciplinary, cultural, and linguistic barriers.
- *Access Research Support Tools*: GoTriple offers tools for visualising research results, web annotation, personalised recommendations, and social networking.

GoTriple currently supports:

- *Automatic Content Classification*: Automatically classifies content in 11 languages at the time of acquisition, including Croatian, English, French, German, Greek, Italian, Polish, Portuguese, Slovenian, Spanish, and Ukrainian.

- *Automatic Annotation*: Utilises the TRIPLE Vocabulary, which includes over 3,300 SSH-related concepts in 12 languages (Croatian, partial Dutch, English, French, Finnish, German, Greek, Italian, Polish, Portuguese, Spanish, and Ukrainian).
- *Multilingual User Interface*: The user interface is localised in several languages, including English, Italian, and French.

Inspired by the Isidore search engine introduced in the previous chapter, the platform serves as the discovery service of OPERAS, a research infrastructure that supports open scholarly communication in SSH within the European Research Area. Additionally, GoTriple is one of the services of the European Open Science Cloud (EOSC), dedicated specifically to the SSH communities.

GoTriple is the main outcome of the European research project TRIPLE (Transforming Research through Innovative Practices for Linked Interdisciplinary Exploration), funded under the Horizon 2020 program and active from October 2019 to March 2023.

3.1.1.1 GoTriple Platform Architecture

The GoTriple platform is organised on two levels: a Core Pipeline and a Discovery Platform. The Core Pipeline ingests, processes, and enriches data imported from aggregators, while the Discovery Platform allows end users to interact with the data through an interface.

The data acquisition process in GoTriple involves two main actors: providers and aggregators. The distinction between these roles is functional and based on specific interactions with the GoTriple platform. Aggregators provide metadata collected from their sources (the providers). GoTriple collects data from both aggregators (such as Isidore, BASE, DOAJ, OpenAIRE, EconStor) and providers (such as OpenEdition, Biblioteka Nauki, ETK, ZRC Sazu, University of Coimbra, Clarin, Cessda, etc.).

3.1.1.2 Data Collection and Normalisation Strategies

The architecture of GoTriple, collecting data from such a wide range of aggregators and providers, necessitated significant efforts to ensure the imported data is homogeneous and easily usable. To address these challenges, various data aggregation and normalisation strategies were implemented throughout the project. These strategies can be distinguished as follows:

- *Metadata Collection*: Metadata is collected from various sources. When available, widely known and used standards, such as OAI-PMH, are reused. This approach allows working

with metadata described by the same schemas, reducing the workload in metadata mapping phases.

- *Metadata Normalisation*: Once collected, the metadata undergoes normalisation processes to standardise different formats and standards. This process includes standardising key attributes such as publication dates, language codes, keywords, document types, licenses, access rights, and author names.
- *Data Mapping*: The normalised metadata is then mapped to the TRIPLE data model.
- *Semantic Enrichment*: The mapped data is further semantically enriched to enhance its discoverability and reuse. This enrichment includes services such as language recognition, automatic translation, automatic classification, and content annotation.
- *Definition of Controlled Vocabularies*: To further improve data consistency, controlled vocabularies were defined for certain attributes. This approach was adopted to standardise and enhance data quality within the platform.

3.1.1.3 Data Model and Ontology

The aforementioned strategies are summarised and described in one of the deliverables produced during the development phases. The document also defines the data model adopted by GoTriple. The purpose of this data model is twofold. Firstly, it defines the target model for the mapping phase. Once a resource is retrieved and normalised, it is aligned with the GoTriple data model. Secondly, it explicitly defines a model that enables semantic interoperability. The primary goal of this work is to produce an ontology aligned with the latest semantic standards, particularly those already existing and widely used within the research communities of the SSH and the ecosystem of research material aggregators.

In the deliverable, the data model is presented as an initial proposal for modelling. Therefore, it was necessary to extend and enrich it. For this reason, the work could not disregard reworking the choices already made within this document. The following section extensively analyses the document, particularly referencing the data model, highlighting its limitations and inconsistencies, which will be resolved in the subsequent sections of this thesis.

3.1.2 Report on Data Enrichment

The D2.5 Report On Data Enrichment (De Santis, 2022) presents the data enrichment strategies used in the TRIPLE project. Specifically, it refers to the Core Pipeline, known as SCRE, through which metadata related to publications and projects in the Social Sciences and Humanities are

automatically collected, mapped to the TRIPLE data model, curated, enriched, and finally saved in the GoTriple platform indexes.

The document begins by illustrating how SCRE imports publication metadata from OAI-PMH endpoints, OpenAIRE data dumps, and Isidore. This reflects the content integration strategies planned in the project. On one hand, OAI-PMH represents a well-established and widely recognized standard for content harvesting; many data providers, especially smaller ones, support it, facilitating their inclusion in GoTriple. On the other hand, support for OpenAIRE and Isidore addresses the need to collect data from large aggregators, a strategy that has enabled GoTriple to quickly present a significant number of publications in its index (over 16 million at the time of writing).

Subsequently, the document describes the normalisation strategies applied to the acquired metadata. By analysing the initial batches of acquired data, rules were defined to normalise and clean attributes related to: publication date, language codes, keywords, document types, licenses, access rights, and author names. The document also presents the definition of controlled vocabularies for some of these attributes.

It then explains the enrichment services, including language recognition, translation, automatic classification, and annotation. Services for detecting duplicate publications and author disambiguation are also discussed, followed by a presentation on the acquisition and processing of project metadata. The document concludes with some final considerations on the data enrichment process, including the difficulties encountered and resolved.

3.1.3 Triple Data Model

Before analysing in detail the contents that need to be considered for ontology development, we will briefly outline them in this section. The contents are concentrated in three main areas: the data model defined for mapping data from external aggregators; the normalisation processes and the metadata they impact; and the issues of duplicate research artefacts and author recognition.

The Triple data model is composed of three parts in the deliverable. The first data model, presented in the table below, pertains to the connectors: Isidore, OpenAIRE, and OAI-PMH. The primary purpose of this data model is to define a data structure to which all metadata retrieved via the connectors can be mapped.

Field	Description	Class
Doi	It includes the valid DOI of the document, if existing.	schema:identifier
Identifier	List of the document's original identifiers	schema:identifier
Title	Title of the document	schema:headline

Abstract	Abstract or description of the document	schema:abstract
Author	Authors of the document: list of authors' names and identifiers (of the Profile Elasticsearch index).	schema:author
Contributor	List of contributors	schema:contributor
Document type	Type of the document	schema:additionalType
Keywords	List of producer's keywords	schema:keywords
Subject (MORESS categories)	List of MORESS Categories (TRIPLE's disciplines)	sioc:topic
TRIPLE thesaurus	TRIPLE thesaurus entries	schema:knowsAbout
Language	Language of the content	schema:inLanguage
Publication date	Date of publication or creation	schema:datePublished
Publisher	List of publishers	schema:publisher
Aggregator	Aggregator of the document (eg: Isidore)	schema:provider
Primary producer	Primary producer of the publication (eg: HALSHS)	schema:producer
License	A license or a type of license	Schema:license
Access Rights	Information on access status	schema:conditionsOfAccess
URL of full document	URL of the full text version of the article	schema:url
URL of the landing page	URL of the landing page of the document.	schema:mainEntityOfPage
Sources information	Source (free text) from: dc:source, dcterms:source (e.g. journal issue)	schema:mentions
Sources URL	Source (HTTP) from: dc:source, dcterms:source (e.g. URL from a publishing platform)	schema:isBasedOnURL
Spatial location of dataset	Content's spatial location of collection (list)	schema:spatialCoverage
Temporal period of dataset	Content's temporal period of collection (list)	schema:temporalCoverage
Format	File format (multiple values)	schema:encodingFormat
Funding reference	List of projects' identifiers (of the Project Elasticsearch index).	schema:funder

Table 1: Triple data model.

The second data model proposed pertains to "profiles." In the deliverable, "profile" refers to the Elasticsearch index, a widely used search engine that stores all records of authors extracted from research artefacts retrieved by connectors. The following table indicates which metadata are

considered essential for describing the "profile" entity. Unlike other tables, no specific mapping to Schema.org is provided in this case.

Field	Description
id	the identifier of the author on the Elasticsearch index, which is automatically calculated. It is based on the normalised name of the author (e.g. spaces are replaced by “_”, accented letters are replaced by their unaccented correspondents, etc) merged with a random string of 21 characters created by the Nano ID library. For example: “César De Santis” can have as id “cesar_de_santis_l6FfvrDgWB1xpW8Ve9U5I”
fullname	The complete name of the author as received in the publication
AKA	This field is only present for duplicates: it contains the id of the author recognised as “original”
author_of	The list of the documents IDs in the Publications Elasticsearch index which are attributed to the author

Table 2: data model of Triple profiles.

Finally, the last data model pertains to projects. GoTriple also provides the capability to track and save all active projects in the Social Sciences and Humanities sector. The following table, similar to the first data model, includes the alignment with the expected classes from Schema.org.

Field	Description	Class
Identifier	Official identifier of the project	schema:identifier
Name	Name of the project	schema:name
Alternate name	Acronym or other name(s) of the project	schema:alternateName
Description	Project description and objectives	schema:description
Start date	Start date of the project	schema:startDate
End date	End date of the project	schema:endDate
Organization	Coordinating entity (eg: CNRS-HN)	schema:organization
Funder	Funder of the project (eg: European Commission)	schema:funder
Funding type	Type of grant (eg: H2020)	schema:fundingScheme
Crowdfunding information	Crowdfunding or agency (can be empty)	schema:sponsor
Keywords		schema:keywords
Subject (MORESS categories)	MORESS	sioc:topic

TRIPLE thesaurus	TRIPLE thesaurus	schema:knowsAbout
URL of the project		schema:URL

Table 3: data model of Triple projects.

Although these three tables in the deliverable are in different sections, it is important to relate them back to the data model presented at the beginning of the deliverable, as they will be necessary during the modelling phase. GoTriple not only models research artefacts but also needs to represent and describe the entities of "profiles" and projects.

Having summarised the state of the art of the deliverable regarding the data model, we can move on to analysing the normalisation processes. Reflecting on these processes is crucial. The fields involved in these procedures may require different modelling approaches depending on the types of transformations they undergo or the types of values they are expected to be associated with. For example, if the normalisation process involves maintaining only a certain range of values for a field retrieved from connectors as free text, the ontology must be modelled according to this restriction. Thus, a field that was previously free text will only be associable with a certain range of values established within the GoTriple ontology.

The fields involved in the normalisation processes are as follows:

- *Language*: the language in which the research artefacts are written
- *Keywords*: keywords associated with the research artefacts, providing information related to the content of the research artefacts
- *Document types*: types associated with the research artefacts
- *Licenses*: licenses associated with the research artefacts
- *Access Rights*: access conditions associated with the research artefacts
- *Author names*: names of people or organisations associated with the research artefacts

The normalisation processes perform one of the following operations on the fields mentioned above:

1. *Normalisation with controlled vocabulary*: After collecting the most frequent values for each metadata subject to normalisation, controlled lists of terms were created. Metadata values are mapped to these controlled lists of terms.

2. *Textual normalisation*: Text fields with character irregularities are normalised according to a set of heuristics.
3. *Discarding*: A value that cannot be mapped to a known or accepted term by GoTriple is considered "discarded." This prevents the value from being used for search purposes within the GoTriple platform.

After reporting the data model and defining which normalisation processes affect which fields, it remains to summarise the issue of research artefact duplication and author disambiguation.

During retrieval, it may happen that the same research artefact exists in more than one connector. However, from a search perspective, it is undesirable for the researcher/user to see duplicate documents or multiple "versions" of the same document. To solve this problem, the research artefact ingestion system is designed to aggregate documents considered "duplicates" into a single container of documents, referred to as a "cluster" in the deliverable and GoTriple architecture. A cluster, therefore, is a set of documents considered duplicates and that can be traced back to a single version of a document. The determining factors for considering a document a duplicate are as follows:

- DOI (if present)
- Title
- Authors
- Year of publication

Another issue considered by GoTriple is the disambiguation of authors. Authors corresponding to each research artefact are extracted, and a "profile" is created for each author. As explained earlier, a profile should not be confused with a user account. It is a temporary entity created to simplify data normalisation. Once profiles are generated within the Elasticsearch "profile" index, the system attempts to match the created profiles to existing profiles. Therefore, at the end of the ingestion and recognition process, multiple profiles may be matched to a single entity: a person or an organisation. This task is necessary because, in the absence of metadata that can uniquely distinguish one person or organisation from another, there is a risk that cases of homonymy or different spellings of the same name may lead to incorrect matches between authors identified by the platform and actual existing people or organisations.

Summarising the points discussed above, we can proceed to their analysis. The results of the analysis should be considered preliminary to the creation of the ontology. This analysis will define the boundaries within which to operate during the modelling phase.

3.2 Output of the Analysis

In this section, we begin by analysing the problem of normalisation, followed by the issue of publication duplication and author disambiguation, and finally, a detailed analysis of the GoTriple data model.

Normalisation processes necessitate considering two aspects at the modelling level:

1. *Creation of Controlled Vocabularies*: "Controlled lists of terms" can be modelled as controlled vocabularies. The values already collected by the developers will form the vocabulary's term set. Specifically, the vocabularies will cover the following entities: licenses, access conditions, document types, languages, and disciplines.
2. *Discarded Entities*: Keywords and author names should have the ability to be marked as "discarded."

Regarding the analysis of duplication problems, it is essential to represent this dual level of research artefacts. A "raw" research artefact, retrieved through a connector from a data provider, should be considered on a different level than the one actually made available by the GoTriple platform to researchers. An example of this functionality, as also reported in the deliverable, is the approach proposed by Google Scholar. When a user performs a search, they can see various relevant results for the entered query on the screen. Each displayed article may present multiple versions linked to the same work.

Thus, the levels to be considered in the modelling of a research artefact are as follows:

- *Raw Research Artefact*: The research artefact returned by the connectors.
- *Cluster of Research Artefacts*: A group of research artefacts defined as duplicates or, in any case, versions of the same research artefact.
- *Unique Research Artefact*: The research artefact obtained after clustering research artefacts recognized as its versions or duplicates.

A similar approach should be applied for author disambiguation. An author should be distinguished as a "Profile," meaning an author extracted directly from the research artefact under analysis, and as a real person or organisation. Consequently, a person or organisation can be linked to multiple

profiles. In the data model, this issue is captured by the field AKA (Also Known As), which connects the profile to other profiles identified as corresponding to the same person.

Finally, we move on to the analysis of the data model. This step is crucial because the proposals within this data model should be considered as the starting point for modelling the GoTriple ontology. The goal in this section is to understand what can be reused in the modelling phase. Specifically, any proposals that would lead to an inconsistency in the ontology will be discarded. The first element that emerges is the choice of the reference ontology, Schema.org. Therefore, for development purposes, Schema.org, where possible, is required as the reference for modelling the GoTriple data model.

In the deliverable's table, only the classes aligned with the existing fields in the GoTriple architecture are reported. However, some of these entities are not modelled in Schema.org as classes. Additionally, the modelling of Schema.org differs from that traced in OWL. While OWL distinguishes between classes, individuals, object properties, and data properties, Schema.org only distinguishes between classes, properties, and individuals. Before proceeding with modelling, it is essential to reconcile Schema.org entities with their types and align them with OWL. This step is fundamental, considering that OWL will be the chosen modelling language.

In the following section, two groups of tables will be presented: the first will clarify the alignment of the complete data model (including the base data model, profile, and projects) to OWL. It will consist of the following columns: the fields considered, their declaration in GoTriple, their declaration in Schema.org, and their declaration in OWL. The second group of tables will aim to categorise the types of fields to be modelled in the ontology development phase. This latter table will be useful for identifying existing ontologies that model the same type of situation.

Before analysing the results, a small additional note is necessary. It should be noted that the TRIPLE thesaurus, format, and subject classes are not included in the tables. The first two were excluded from modelling in general due to their lack of interest. Subject, with the `sioc:topic` property, is already aligned with OWL as an object property.

It starts with the alignment tables:

Field	Schema.org URI	Schema.org Type	OWL Type
Identifier	schema:identifier	Property	Object property or Data property
Abstract	schema:abstract	Property	Data Property
DOI	schema:identifier	Property	Object property or Data property

Title	schema:headline	Property	Data Property
Author	schema:author	Property	Object property
Contributor	schema:contributor	Property	Object property
Document type	schema:additionalType	Property	Object property
Keywords	schema:keywords	Property	Object property or Data property
Language	schema:inLanguage	Property	Object property or Data property
Publication date	schema:datePublished	Property	Object property
Publisher	schema:publisher	Property	Object property
Aggregator	schema:provider	Property	Object property
Primary Producer	schema:producer	Property	Object property
License	schema:license	Property	Object property or Data property
Access rights	schema:conditionsOf Access	Property	Object property or Data property
Url of full document	schema:url	Property	Data property
Url of the landing page	schema:mainEntityOf Page	Property	Object property
Sources information	schema:mentions	Property	Object property
Sources URL	schema:isBasedOnUrl	Property	Object property
Spatial location of dataset	schema:spatialCoverage	Property	Object property or Data property
Temporal location of dataset	schema:temporalCoverage	Property	Object property or Data property
Format	schema:encodingFormat	Property	Object property or Data property
Funding reference	schema:funder	Property	Object property

Table 4: Alignment of the Triple data model with OWL.

Field	Schema.org URI	Schema.org Type	OWL Type
-------	----------------	-----------------	----------

Id	/	/	Object property or Data property
fullname	/	/	Object property or Data property
AKA	/	/	Object property
author_of	/	/	Object property

Table 5: Alignment of the Triple Profile data model with OWL.

Field	Schema.org URI	Schema.org Type	OWL Type
Identifier	schema:identifier	Property	Object property or Data property
Name	schema:name	Property	Data property
Alternate name	schema:alternateName	Property	Data property
Description	schema:description	Property	Data property
Start date	schema:startDate	Property	Object property
End date	schema:endDate	Property	Object property
Funder	schema:funder	Property	Object property
Organization	schema:organization	Property	Object property
Funding Type	schema:fundingScheme	Property	Object property
Crowdfunding information	schema:sponsor	Property	Object property
Keywords	schema:keywords	Property	Object property or Data property
URL of the project	schema:URL	Property	Object property or Data property

Table 6: Alignment of the Triple Project data model with OWL.

The division tables by categories are proceeded:

Field	Category
Identifier DOI Sources URL	Identifier

Url of full document Url of the landing page Sources information	
Abstract Title	Content metadata
Author Contributor Aggregator Publisher Primary Producer	Roles
Publication date	Temporal collocation
Language	Language
Keywords Temporal location of dataset Spatial location of dataset	Subject coverage
Document type License Subject Access rights	Controlled vocabulary

Table 7: Categories Triple data model fields.

Field	Category
Id	Identifier
fullname	Content metadata
AKA	Other Profile connection
Author of	Connection with research artefacts

Table 8: Categories Triple profile fields.

Field	Category
Identifier Url of the project	Identifier
Name Description Alternate Name	Content metadata
Funder	Roles

Sponsor Coordinating entity	
Grant	Temporal collocation
Keywords	Subject coverage
Subject	Controlled vocabulary

Table 9: Categories Triple project fields.

The tables reveal several important aspects that can be summarised as follows:

1. *Three Key Entities*: The entities to be considered during the modelling phase are research artefacts, projects, and profiles.
2. *Class and Property Considerations*: The classes defined as fields in the Triple data model should be considered as properties in Schema.org and as object/data properties in OWL. All these properties need to be taken into account during the modelling phase.
3. *Macro Areas for Fields*: The fields can be distributed into the following macro areas:
 - *Identifier*: Fields designed to provide unique identifiers for the resource in question.
 - *Controlled Vocabularies*: As noted in the analysis of normalisation processes, some fields appear as "controlled lists of terms." Therefore, fields under this macro area should be modelled as classes linked to controlled vocabularies.
 - *Subject*: Fields aimed at defining the coverage of the article's content.
 - *Textual Fields*: Fields that report the content of the analysed resource.
 - *Temporal Coverage*: Fields intended to place the analysed object in a temporal context.
 - *Roles*: Individuals or organisations involved in various capacities in the creation of the analysed resource.
 - *Language*: Fields that identify the language of the analysed resource.

From the analysis phase, several fundamental points emerge that need to be considered for the ontology development. The most important of these is the initial alignment between Schema.org, the Triple data model, and OWL, the language that will be used for defining the ontology.

3.3 Ontology Design and Development

This section focuses on the methodology used for developing the ontology, called Simplified Agile Methodology for Ontology Development (SAMOD, <https://essepuntato.it/samod/>) (Peroni 2016), along with a series of supporting applications used during the development process.

Over the years, many methods and methodologies have been developed and used for engineering ontological models. In order to be successfully applied in enterprise contexts, which are usually characterised by high levels of complexity, most of these methodologies have been built on a robust, framed architecture, such as the Cyc method (Lenat et al. 1990), Uschold and King's method (Uschold & King 1995), Gruninger and Fox's methodology (Gruninger & Fox 1995), METHONTOLOGY (Fernández-López et al. 1997) and the Neon Methodology (Suárez-Figueroa et al. 2012), among many others. Although effective, these approaches imply major cognitive work and, as a result, hinder easily viable changes or reuses of the ontologies that have been created. In consideration of this, during the last couple of decades, a series of light-weight methodologies, gathered under the umbrella term Agile, have been developed and successfully applied whenever frequently changing requirements and limited amounts of entities were the norm and not an exception. One of the first agile methodologies introduced in the Ontology Engineering domain was RapidOWL (Auer & Herre 2007), based on the idea of an incremental, evolutionary modelling of knowledge iteratively built in a close collaboration between users, domain experts and knowledge engineers. Another exemplary methodology is represented by eXtreme Design (Presutti et al. 2009), which exploits ontology design patterns to resolve modelling issues and competency questions to define the functional requirements of the model. A more recent example is UPON Lite (De Nicola & Missikoff 2016), a simple, agile ontology engineering method derived from the highly structured UPON methodology (De Nicola et al. 2005) and organised as an ordered set of steps, each releasing a self-contained artefact readily available to end users.

Recently, Peroni (Peroni, 2016) proposed a new agile methodology for developing ontologies called Simplified Agile Methodology for Ontology Development (SAMOD), which has been used to develop the Triple from the beginning. SAMOD, ontologies that have been reused and supplementary tools are introduced in the following subsections.

3.3.1 Simplified Agile Methodology for Ontology

3.3.1.1 Development

The Simplified Agile Methodology for Ontology Development (SAMOD, <https://essepuntato.it/samod/>) introduced by Peroni in 2016 is a methodology designed for ontology development, inspired by the Test-Driven Development in Software Engineering and other agile methodologies such as eXtreme Design (XD). SAMOD is structured around a three-step, iterative process focused on producing well-developed, documented, and human-readable ontologies using exemplar data.

The methodology proceeds as follows, with each step culminating in a "milestone" — the formal implementation of the ontology in its current state supplemented by resources like glossaries, diagrams, and query examples:

1. **Test Case Definition:** Ontology engineers (OEs), in collaboration with domain experts (DEs), gather domain data from a Motivating Scenario (MS) that provides an initial informal semantics for intended concepts and relations. Subsequently, a list of Informal Competency Questions (CQ) and a Glossary of Terms (GoT) are developed to refine entity and property identification within the ontology. Utilising these resources, the OEs create a preliminary model (modelet, TBox) formalising these entities and properties, complemented by a dataset (ABox) illustrating instances from the MS. Formal Competency Questions (SQ) are then crafted by translating CQ into SPARQL queries. A new test case, incorporating these resources, is developed. If it successfully passes the model, data, and query tests, a milestone is achieved, leading to the next step.
2. **Merging:** This step involves integrating the modelet from the first step with the previously released model by merging their axioms, classes, and properties, and collapsing semantically identical elements. All existing test cases are updated to reflect these changes in the TBox, ABox, and SQ. Successful passage of the model, data, and query tests for each test case results in another milestone, after which the merged model becomes the current model for the subsequent step.
3. **Refactoring:** The final step requires OEs to refine the current model shared across all test cases, along with the ABox and SQ for each. Refactoring is guided by best practices such as reusing existing models, adding descriptive natural language labels, and enhancing the

model to maximise the inferable information using OWL 2 DL technologies. Following updates and validations of all test cases, if the refactored model passes all tests, a milestone is finalised, marking the end of the current iteration. The process then continues to the next iteration as necessary.

SAMOD has been employed since its inception for developing the Triple ontology, involving an OE (the author) with assistance from his supervisor and a DE (the assistant supervisor).

3.3.1.2 Reused Model

In the third step of every iteration, SAMOD suggests importing other models into the ontology whenever possible to maximise reusability in other contexts. The reused models are as follows:

- Schema.org (Schema) (<https://schema.org/>): Schema.org vocabulary can be used with many different encodings, including RDFa, Microdata and JSON-LD. These vocabularies cover entities, relationships between entities and actions, and can easily be extended through a well-documented extension model. Over 10 million sites use Schema.org to markup their web pages and email messages.
- Semantically-Interlinked Online Communities (SIOC) (<http://www.w3.org/submissions/sioc-spec/>): this ontology provides the main concepts and properties required to describe information from online communities (e.g., message boards, wikis, weblogs, etc.) on the Semantic Web.
- Dublin Core Metadata Terms (DCTerms) (<http://purl.org/dc/terms/>): This specification implements all the metadata terms maintained by the Dublin Core Metadata Initiative, including properties, vocabulary encoding schemes, syntax encoding schemes, and classes.
- FRBR-aligned Bibliographic Ontology (FaBiO) (<http://purl.org/spar/fabio>) (Peroni & Shotton, 2012): Based on the FRBR model, this ontology is used for describing bibliographical entities in all their various essences.
- Friend Of A Friend (FOAF) (<http://xmlns.com/foaf/0.1>) (Brickley & Miller, 2007): A vocabulary for describing people and their relations with other people, documents, and other information objects.
- Publishing Role Ontology (PRO) (<http://purl.org/spar/pro/>) (Peroni, Shotton & Vitali, 2012): an ontology for the characterisation of the roles of agents – people, corporate bodies and

computational agents in the publication process. These agents can be, e.g. authors, editors, reviewers, publishers or librarians.

- Expression of Core FRBR Concepts in RDF (FRBRcore) (<http://purl.org/vocab/frbr/core>): An RDF vocabulary incorporating the basic concepts and relations described in the IFLA report on the Functional Requirements for Bibliographic Records (FRBR).
- Literal Reification (<http://purl.org/spar/literal>) (Gangemi et al., 2010): An ontology pattern allowing certain literals to be modelled as individuals of a class, so they can be used as proper subjects or objects of RDF statements within an ontology.
- Time Interval (TI) (<http://www.ontologydesignpatterns.org/cp/owl/timeinterval.owl>) (Gangemi & Presutti, 2009): An ontology pattern extracted from DOLCE+DnS UltraLite (DUL, <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl>). It enables the description of periods of time, each characterised by a starting date and an ending date.
- Time-indexed Value in Context (TVC) (<http://purl.org/spar/tvc>): An extension of the Time-indexed Situation pattern (TISit, <http://www.ontologydesignpatterns.org/cp/owl/timeindexedsituation.owl>). This ontology pattern describes scenarios characterised by four main elements: an entity having a value, the value held by the entity, a time interval during which the entity has that value, and a context in which the act of having that value makes sense. It allows the description of situations in which entities have values during a particular time and within a particular context.

3.3.1.3 Tools

The Live OWL Documentation Environment (LODE, <http://www.essepuntato.it/lode>) (Peroni et al., 2012) is an open-source service that automatically extracts classes, object properties, data properties, named individuals, annotation properties, general axioms, and namespace declarations from an OWL ontology. It then renders them as ordered lists, together with their textual definitions, in a human-readable HTML page designed for browsing and navigation. During the development of Triple, LODE was used to produce the HTML documentation of the ontology by extracting the labels, comments, and provenance information of the ontology elements.

The Graphical Framework for OWL Ontologies (Graffoo, <http://www.essepuntato.it/graffoo>) (Falco et al., 2014) is an open-source tool used to present classes, object properties, data properties, individuals, general axioms, namespace declarations, and restrictions within OWL ontologies as

user-friendly diagrams. A Graffoo diagram illustrates the logical relationships between elements of an ontology in an easy-to-understand format. During the development of Triple, Graffoo was utilised to create various modelets and all the diagrams of the ontology.

Protégé (<https://protege.stanford.edu/>) (Noy et al., 2001) is an open-source ontology editor developed at Stanford University. It provides a graphical user interface, deductive classifiers, and OWL 2 DL reasoning engines (e.g., HermiT and Pellet) to validate the consistency of an ontology and infer new knowledge from it. Protégé was used multiple times in each iteration for testing the consistency of the TBox and the ABox, the merged model, and the refactored model of each iteration.

Apache Jena Fuseki is a SPARQL 1.1 (<http://www.w3.org/TR/sparql11-overview/>) server with a web interface, backed by the Apache Jena TBD RDF triple store. It provides the SPARQL 1.1 protocols for query and update, as well as the SPARQL Graph Store protocol. Fuseki was used multiple times in each iteration as a query engine for querying the ABox and the refactored ABoxes of each iteration to test the effectiveness of the formal competency questions and to address the particular requirements they expressed.

Chapter 4: Triple ontology

This chapter analyses the core elements of the ontology. After a brief review of the namespaces adopted and the criteria used to modularize the ontology, it proceeds to detail the entities and relationships found in Triple ontology. Next, the chapter delves into which ontologies were reused and where they were applied. The last section illustrates a series of examples that outline possible fields of application of the Triple ontology.

4.1 GoTriple Ontology Structure & Components

This section introduces the main ontological elements defined in Triple. The base URI for Triple is <https://gotriple.eu/ontology/triple/>. The preferred prefix is triple. The repository with all materials <https://github.com/AlessandroBertozzi/TRIPLE-ontology/tree/main>. The model has been developed by following a middle-out approach, based on the domain expert's needs and the use of simulated scenarios provided with toy data that have been structured around them. The prefixes and relative base URIs of the models which Triple reuses are listed below.

Prefix	Base URI
triple	https://gotriple.eu/ontology/triple/
schema	http://schema.org/
foaf	http://xmlns.com/foaf/0.1/
skos	http://www.w3.org/2004/02/skos/core#
pro	http://purl.org/spar/pro/
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
owl	http://www.w3.org/2002/07/owl#
xsd	http://www.w3.org/2001/XMLSchema#
rdfs	http://www.w3.org/2000/01/rdf-schema#
fabio	http://purl.org/spar/fabio
dc	http://purl.org/dc/
sioc	http://rdfs.org/sioc/ns#
dcterms	http://purl.org/dc/terms/
datacite	http://purl.org/spar/datacite

ti	http://www.ontologydesignpatterns.org/cp/owl/ti-meinterval.owl#
tvc	http://www.essepuntato.it/2012/04/tvc/
collectionentity	http://www.ontologydesignpatterns.org/cp/owl/collectionentity.owl

Table 10: Triple Ontology prefix.

4.1.1 Namespaces

Ontology modularization is a technique used in ontology engineering to divide a complex ontology into smaller, more manageable modules. This approach enhances the clarity, reusability, and maintainability of ontological structures. Modular ontologies are easier to develop, test, and extend, allowing for more efficient management of large and intricate knowledge domains. Modularizing an ontology offers several key advantages:

- **Manageability:** Breaking down an ontology into smaller modules makes it easier to manage. Each module can be developed, updated, and maintained independently, reducing the complexity of the overall ontology.
- **Reusability:** Modules can be reused across different ontologies or projects. This promotes the standardisation of terms and definitions, facilitating interoperability between systems.
- **Scalability:** New modules can be added as needed, allowing the ontology to grow and adapt to new requirements without disrupting existing structures.
- **Clarity and Documentation:** Modular ontologies are easier to understand and document. Clear namespace divisions help users and developers navigate the ontology and understand the relationships between different concepts.

The GoTriple ontology adopts a modularization strategy to enhance its manageability and scalability. Each module is assigned a distinct namespace that clearly indicates its purpose and scope. The following sections define the namespaces used in the GoTriple ontology and provide a brief description of each module.

Name	Namespace	Description
Triple	https://gotriple.eu/ontology/triple/	The main ontology

Document	https://gotriple.eu/ontology/triple/document	This module is designed to represent all documents within the GoTriple platform, encompassing various research artefacts. It includes terms related to publications, reports, datasets, and other scholarly outputs.
Profile	https://gotriple.eu/ontology/triple/profile	The profiles module represents all author profiles extracted from publications within GoTriple or declared by users when registering to the platform.
Project	https://gotriple.eu/ontology/triple/project	This module is used to represent research projects within GoTriple. It includes terms related to project titles, descriptions, participants, funding sources, and outcomes.
GoTriple Vocabulary Ontology (GTVO)	https://gotriple.eu/ontology/triple/gtvo	The GTVO (GoTriple Vocabulary Ontology) schema module forms the basis of all controlled vocabularies in GoTriple. It includes general terms and structure for controlled vocabularies used in the platform.
Disciplines	https://gotriple.eu/ontology/triple/gtvo/disciplines	This vocabulary module defines terms related to academic disciplines. It is based on the

		GTVO schema and includes a controlled list of disciplines to categorise research outputs.
Conditions of Access Vocabulary	https://gotriple.eu/ontology/triple/gtvo/conditions_of_access	This module includes terms related to the conditions under which research artefacts can be accessed. It is based on the GTVO schema and helps specify access rights and restrictions.
Licences Vocabulary	https://gotriple.eu/ontology/triple/gtvo/licenses	The licences vocabulary module defines terms for different types of licences applicable to research artefacts. It is based on the GTVO schema and includes common licensing terms and conditions.
Resources Types Vocabulary	https://gotriple.eu/ontology/triple/gtvo/resources_types	This module includes terms describing various types of resources stored in GoTriple. It is based on the GTVO schema and helps categorise resources such as datasets, publications, and software.

Table 11: Triple Ontology Namespaces.

4.1.2 FRBR alignment with FaBiO

In the modelling of GoTriple, one of the central entities is the document. As will be discussed in detail later, the document encompasses all research artefacts collected within GoTriple. The fundamental goal of GoTriple is to make these research artefacts available to a potential researcher in the social sciences and humanities (SSH) through its platform. As anticipated in Chapter 3, the

document must be defined on different levels of conceptualization to be correctly represented. There is a significant difference between the documents collected from external data providers and the documents served to researchers. Additionally, to ensure compatibility with external models, it was necessary to develop a more complex representation than a simple "documents" class that would encompass all research artefacts.

Given these considerations, the decision was made to design Triple based on the Functional Requirements for Bibliographic Records (FRBR) standard, a comprehensive and flexible model proposed by the International Federation of Library Associations (IFLA) for the representation of bibliographic resources and metadata. FRBR is adaptable to various implementations and applicable to both physical and digital resources. It conceptualises each resource from four interconnected perspectives, categorised as follows:

- Work: This represents the abstract essence of a resource, independent of any physical form. A Work is realised through one or more Expressions.
- Expression: This denotes the form a Work takes when materialised in terms of content. Each Expression is a realisation of a single Work and is embodied in one or more Manifestations.
- Manifestation: This signifies a specific physical embodiment of an Expression, in a particular format. Each Manifestation embodies one or more Expressions and is exemplified by one or more Items.
- Item: This refers to a single, tangible instance of a Manifestation. Each Item exemplifies one specific Manifestation.

These concepts are structured in a relational hierarchy, forming a seamless flow from Work to Item and vice versa. This framework enables a comprehensive perspective on resources, addressing various levels of abstraction and resolving semantic ambiguities related to human-made objects by delineating distinct yet interconnected concepts. It facilitates a more expressive, precise, and dynamic description of artefacts and their interrelations.

Despite its strengths, FRBR has certain limitations. Although the definitions of its concepts are straightforward, the terminology (Work, Expression, Manifestation, and Item) can be confusing for the average user. According to FRBR, a complete description of an object (such as a document) necessitates consideration of all four levels, which can be challenging for users who intuitively expect the concept to exist at a single level.

To mitigate this issue while maintaining FRBR's expressiveness, it is practical to position an object at the most appropriate level relative to the given scenario. This strategy has been systematically applied in the FRBR-aligned Bibliographic Ontology (FaBiO), with which Triple is aligned. FaBiO, an ontology focused on entities that are published, textual, and/or cited in bibliographic references, defines its own set of entities as subclasses of the original FRBR entities, leveraging the FRBRcore RDF vocabulary.

To adopt this approach and align with the case study scenarios, Triple reuses the interpretations of FRBR entities from FaBiO as the superclass for one of its entities: *document*. As outlined in the metadata analysis (see Chapter 3), data providers often return the same document in various situations. This does not necessarily mean duplicates, but rather the same document in different formats. Referring to the description of `fabio:Manifestation`, the same version of a document can appear under different manifestations from different data providers. Therefore, duplicates in GoTriple are not merely repeated documents but different manifestations of the same document.

For this reason, the document in GoTriple can be viewed as a subclass of `fabio:Expression`. In the Triple ontology, however, the document can be considered aligned with the document defined by FOAF, `foaf:Document`. Formalising `foaf:Document` as a subclass of `fabio:Expression` would result in a very strong assertion, making it incompatible with other ontologies that reuse the `foaf:Document` class in different scenarios. To address this, a third specific class in Triple, `triple:Document`, has been created. This class is a subclass of both `foaf:Document` and `fabio:Expression`. Documents collected directly from data providers, including duplicates, will belong to the `fabio:Manifestation` class.

4.1.3 Model entities

This section aims to introduce the main classes of Triple that form its ontological foundation. Specifically, it will describe the classes that represent the primary entities managed by the GoTriple platform (documents, profiles, projects, and controlled vocabularies), the classes necessary to describe these entities (roles, publication dates, online accounts, etc.), and the classes that have been created specifically to describe unique entities within this ontology (clusters, profiles, profile names).

The central entity of the ontology is represented by documents. In GoTriple, the term "documents" technically refers to the Elasticsearch index that stores all research artefacts imported from external data providers. As reiterated throughout this work, GoTriple is a discovery platform. It does not

create resources but aggregates them from various external data providers. The platform's goal is to offer a single access point to all resources under the umbrella of Social Sciences and Humanities (SSH).

More broadly, a research artefact, and thus a document in GoTriple, is any tangible or intangible output produced in the course of academic and scientific research. These artefacts are fundamental elements that document, support, and communicate research results, as well as the data and methodologies used.

The various types of documents that can be encountered in GoTriple are indicated through the controlled vocabulary "document types". The controlled vocabularies will be described later in this section.

To formalise this entity, GoTriple defines the class `triple:Document`. As described in the section on FRBR and FaBio (see FRBR alignment with Fabio), in addition to this class, it was necessary to reuse the classes `foaf:Document` and `fabio:Expression`.

Regarding documents, it is also important to highlight another entity: the "cluster". This term, like documents, originates from technical choices in the platform's definition. A cluster is technically a group of documents considered by the system as "duplicates" of a single document. Often, these documents include not only duplicates but also manifestations of a version of a document (see FRBR alignment with Fabio). To represent this entity, a specific class for the GoTriple ontology was adopted: `triple:Cluster`. The `triple:Cluster` was represented using an Ontology Design Pattern (ODP), specifically the Collection pattern. The `triple:Cluster` class corresponds to the `owl:Collection` class. As members, it enumerates the manifestations contained in `fabio:Manifestation` of a `triple:Document`.

A project within GoTriple is a structured and organised research initiative that involves one or more researchers and aims to generate new knowledge or deepen specific topics in the field of Social Sciences and Humanities. These projects are often funded by academic institutions, government agencies, or private organisations and progress through various stages, from hypothesis formulation to data collection and analysis, culminating in the publication of results. Within Triple, a project is represented by the class `foaf:Project`.

For the publication of a `triple:Document` or the definition of a `foaf:Project`, various individuals or organisations can collaborate, assuming specific roles. To represent these entities, Triple reuses the pattern proposed by the PRO ontology (Peroni & Shotton, 2012) (<http://purl.org/spar/pro/>). In this ontology, individuals and organisations are represented by the classes `foaf:Person` and `foaf:Organization`, respectively. To represent the roles that a

person or an organisation can assume in relation to a document or a project, PRO uses the class `pro:Role`. Individuals and organisations are grouped under a single entity: `foaf:Agent`. An agent is generally defined as someone “who causes a particular effect through their action” (https://www.oed.com/dictionary/agent_n1?tab=factsheet#8694696). In this case the effect is understood to be a role in the publication or management of a project.

In addition to describing who performs the action and what role they assume, the model includes another entity of central importance for modelling: `pro:RoleInTime`. This entity connects a `triple:Document` or a `foaf:Project` to an agent who assumes a specific role in the context of that `triple:Document` or `foaf:Project`. This class can also be temporally restricted, indicating that a specific `foaf:Agent` held a certain `pro:Role` within the context of a `foaf:Project` or a `triple:Document` during a specific time period. Although this temporal restriction was included in the final ontology, it is not necessary as this type of information was not part of the requirements defined with the domain expert.

To better illustrate the function of `pro:RoleInTime`, consider this example: take two academic publications, both having the same author. In this case, we have one `pro:Agent`. However, the roles will differ. In the first publication, the `foaf:Agent` serves as the author, while in the second, they are merely a contributor. Distinguishing in which `triple:Document` the `pro:Agent` held each role is not possible without an intermediary class. This is where `pro:RoleInTime` comes into play. The documentation describes this entity as “A particular situation that describes a role an agent may have, which can be restricted to a particular time interval” (Peroni & Shotton, 2012).

The final aspect to note regarding the `pro:RoleInTime` class is its extension with `triple:Profile`, discussed later in this section.

Another central entity in GoTriple is “profiles.” Similar to `triple:Document`, this term is inherited from the name of the Elasticsearch index that stores this type of information. This index is populated by extracting the names of authors attached to `triple:Document`. In Triple, it was not feasible to simply reuse the `foaf:Person` or `foaf:Organization` class to define this entity. Multiple profiles might correspond to a single `foaf:Person` or a single `foaf:Organization`. The subtle difference is tied to the difficulty of tracing the name and surname of an author extracted from a `triple:Document` back to a unique and real `foaf:Person`. Therefore, a specific class, `triple:Profile`, was defined for this ontology.

A class with a similar role, however, already exists in Triple. Reflecting on the `pro:RoleInTime`

class, it becomes evident that `triple:Profile` merely, as said before, defines an agent in a certain situation with a specific `pro:Role`. The only difference is that it refers to an agent in a `triple:Document` at the level of `fabio:Manifestation`. Hence, `triple:Profile` is defined as a subclass of `pro:RoleInTime`, extending it.

However, the `triple:Profile` class needs further extension. Another element to represent is the agent's name at the level of `fabio:Manifestation` of a `triple:Document`. While at the `fabio:Expression` level, an agent can be identified as a person, at the manifestation level, the same agent might be named differently for editorial reasons. This necessity led to the representation of the agent's name in Triple, more specifically the name of `triple:Profile`. To define this entity, the class `triple:ProfileName` is used.

Thus, the `triple:Profile` class allows for extending the levels of representation of agents. Agents differ, especially in the writing of their names, at the manifestation level of a `triple:Document`.

In Triple, each `foaf:Person` can be associated with an account. Creating an account is crucial for a user who wants to link their identity to the `triple:Document` available on the platform. Often, the available metadata of individual documents about the authors is insufficient to unequivocally associate a document or a project with a real person. Therefore, users can "claim" their articles, assisting the platform's automatic processes in disambiguating authors. In Triple, the account entity is defined by the `foaf:OnlineAccount` class.

A controlled vocabulary is a standardised and organised set of predefined terms and phrases used to ensure consistency and accuracy in the description and classification of information. Specifically, a controlled vocabulary is used to index content, facilitate information search and retrieval, and improve interoperability between different systems and datasets. In Triple, controlled vocabularies are essential for describing the following entities related to `triple:Document` and `foaf:Project`: disciplines, document types, access conditions, and licences. As outlined in the metadata recognition (see 3.1 Outcome of Analysis), controlled vocabularies arise from the need to normalise the variety of expected values for certain metadata. After analysing the most frequent values, they were used to populate the controlled vocabularies mentioned above.

To represent these entities in Triple, the classes provided by SKOS (Simple Knowledge Organization System) were reused. For terms, the `skos:Concept` class was employed, and to define the vocabularies, the `skos:ConceptScheme` class was used.

4.1.4 Model relationships

Once the primary classes that comprise the Triple ontology have been clarified, we can outline the main relationships formalised within the ontology. This section delineates the relationships between the central entities in Triple: vocabulary, project, document, and profile. We will outline the relationships that are uniquely defined within this ontology and reference the external ontologies employed to represent ancillary relationships.

The classes `triple:Document`, `foaf:Project`, and `triple:Profile` in Triple are connected through the definition of `pro:Role`. As described in the previous section (see 4.1.3 Model Entities), the `triple:Profile` can be defined as a subclass of `pro:RoleInTime`. A document defines a relationship with this class at both the `fabio:Expression` and `fabio:Manifestation` levels through the property `pro:isDocumentContextFor`. The `foaf:Project` connects to the representation of roles using the superclass of the property used by documents, `pro:isRelatedToRoleInTime`. The property `pro:isDocumentContextFor` was not reused for `foaf:Project` because it is too specific to documents. Therefore, the more generic superclass was chosen.

Controlled vocabularies are related to both `triple:Document` and `foaf:Project`. The relationships used are as follows: `dc:type` to connect documents to their respective types; `sioc:topic` to connect documents and projects to a research discipline; `schema:conditionsOfAccess` to connect documents to access conditions; and `schema:licence` to connect documents to licences. Each of these relationships links a project or document to a `skos:Concept`, which in turn is linked to a `skos:ConceptScheme` through the relationship `skos:inScheme`.

The specific relationship of Triple that could not be described using properties from other ontologies is `triple:ProfileName`. As described in the previous section (see 4.1.3 Model Entities), each profile may present a name written differently from other documents, even though it always refers to the same `foaf:Person`. For this reason, the subclass of `pro:RoleInTime` needs a relationship to describe the name of a `foaf:Agent` in that temporal role. In Triple, the relationship `triple:ProfileName` is defined and is linked to an `rdfs:Literal` type.

Numerous other relationships are represented in Triple. For these properties, the ontologies reused are SKOS, FOAF, Schema.org, PRO, and DataCite. These ontologies will be described in detail in the next sections of this chapter.

4.1.5 SKOS

The Simple Knowledge Organization System (SKOS) (<https://www.w3.org/TR/skos-reference/>) is a widely-adopted framework for representing knowledge organisation systems such as thesauri, classification schemes, subject heading systems, and taxonomies. It facilitates the expression of these systems as machine-readable data, enabling interoperability between different computer applications and the ability to publish this data on the Web in a standardised format (Sánchez et al., 2009).

According to the official W3C documentation “SKOS simple knowledge organisation system reference” (Alistair & Bechhofer, 2009), the SKOS data model is formally defined as an OWL Full ontology, integrating closely with the standards of the Web Ontology Language (OWL) and Resource Description Framework (RDF). SKOS data are expressed in the form of RDF triples and can be encoded using any RDF syntax, such as RDF/XML or Turtle. This integration ensures that SKOS benefits from the semantic richness and interoperability of OWL and RDF, making it a suitable choice for knowledge representation.

At the core of SKOS is the concept of a "concept scheme," which is essentially a set of concepts organised for a specific purpose. Each concept and concept scheme is identified by a Uniform Resource Identifier (URI), ensuring reference across different contexts and making them part of the World Wide Web.

SKOS allows concepts to be labelled with lexical (UNICODE) strings in multiple languages, enabling multilingual support. Each concept can have a preferred label in each language, as well as alternative and hidden labels.

In addition to URIs, SKOS concepts can be assigned notations—lexical codes that uniquely identify a concept within a particular scheme. Notations serve as a bridge between SKOS concepts and other existing classification systems, such as those used in library catalogues. It supports a variety of documentation properties for annotating concepts with notes, including scope notes, definitions, and editorial notes. This flexibility enables extensive documentation, which is essential for clarifying the meaning and use of concepts. SKOS provides semantic relation properties to link concepts hierarchically or associatively, supporting the creation of complex knowledge structures. Concepts can be grouped into collections, which can be labelled and ordered. This feature is particularly useful for representing node labels in thesauri or for situations where the sequence of concepts conveys additional meaning. SKOS also supports mapping concepts across different concept schemes using hierarchical, associative, close, and exact equivalence links. These mappings are essential to enable interoperability and integration between different knowledge organisation systems.

Thus, SKOS is a Semantic Web language designed to represent controlled structured vocabularies such as thesauri, classification schemes, subject heading systems, and taxonomies. It provides a standardised framework for publishing these vocabularies on the Web, facilitating their integration and application in various resource collections within the Semantic Web. The importance of this ontology lies in its ability to enable interoperability between different knowledge organisation systems, enabling data retrieval and integration services between different collections.

Within the Triple ontology, SKOS plays a fundamental role in the construction of controlled vocabularies. As mentioned in the previous section, there are four controlled vocabularies: licences, conditions of access, document types, and disciplines. All these vocabularies have been built using the GTVO (GoTriple Vocabulary Ontology) ontology, which is also defined within this project.

The construction of GTVO is a strategic choice aimed at isolating the underlying structure of all controlled vocabularies within Triple from the controlled vocabularies themselves. SKOS is ideal for building the GTVO module. As stated earlier, SKOS is designed to model controlled vocabularies, whose concepts can be connected through relationships of varying degrees (exact, close, etc.). In Triple, it is necessary to manage controlled vocabularies and especially to create terms within controlled vocabularies that can be connected to other terms defined in external vocabularies. Therefore, SKOS is ideal for two reasons: it is a well-known and widely used ontology in the semantic web community, and it perfectly covers the use case in question.

To construct GTVO, the following entities and relationships from SKOS were selected:

- `skos:Concept`
- `skos:ConceptScheme`
- `skos:definition`
- `skos:exactMatch`
- `skos:closeMatch`

With these entities, all the requirements of our use case can be met:

- Each concept (`skos:Concept`) can be traced back to a controlled vocabulary (`skos:ConceptScheme`).

- Each concept can be connected to external concepts through a relationship indicating a complete overlap of concepts (`skos:exactMatch`) or a strong relationship that is not sufficient to define them as identical (`skos:closeMatch`).
- Each concept can be connected to labels in various languages, thus supporting multilingualism.

GTVO is integrated with another ontology, DataCite, specifically for modelling the representation of codes related to `skos:Concept`. DataCite will be illustrated in the subsequent sections.

4.1.6 FOAF

FOAF, an acronym for Friend of a Friend, is a project designed to describe the world using simple, web-inspired concepts. According to the official documentation (Brickley & Miller, 2007) The core principle behind FOAF is the representation of entities (people, groups, documents, etc.) and their interrelationships using a set of defined classes and properties. In FOAF, these entities are categorised as classes, while the relationships between them are defined as properties. Essentially, FOAF acts as a dictionary of terms, each term representing either a class or a property. This structured approach allows for the integration of other projects that provide complementary sets of classes and properties, many of which can be linked with those defined by FOAF.

FOAF terms are grouped into broad categories based on their applicability and use cases. This overview focuses on the primary categories, excluding archaic and historical terms, and distinguishes between terms relevant to web-specific contexts and those with universal applicability.

The core classes and properties of FOAF describe fundamental characteristics of people and social groups that are timeless and technology-independent. These terms can be used to represent basic information about individuals across various contexts, including present-day, historical, cultural heritage, and digital library applications. In addition to personal characteristics, FOAF defines classes for Project, Organization, and Group as other types of agents.

Beyond the core terms, FOAF includes a set of terms tailored for describing Internet accounts, address books, and other web-based activities. These terms support the representation of social interactions and identities in the digital realm. Related initiatives in this category include Portable Contacts and the W3C Social Web group.

FOAF originated as the 'RDFWeb' project and has been instrumental in promoting the model of publishing simple factual data through a network of linked RDF documents. FOAF continues to play a crucial role in the Linked Data community, emphasising the integration of factual

information with human-oriented documents, such as videos, books, spreadsheets, and 3D models. Additionally, FOAF includes demonstration terms (e.g., `geekcode`) for educational purposes and technical utility terms (e.g., `focus`, `LabelProperty`) that facilitate broader information-linking efforts. This pragmatic, dictionary-based design of FOAF reflects its central aim: to link networks of information with networks of people.

In Triple, FOAF has been reused to define the most important entities within the namespaces of document and project. The entity `triple:Document` is modelled as a subclass of `foaf:Document` and `fabio:Expression`. The project entity is modelled by reusing `foaf:Project`. For `triple:Profile`, `foaf:OnlineAccount` is used. Within PRO, described later, FOAF is reused to define `foaf:Person`, `foaf:Organization`, and `foaf:Agent`.

The choice to use FOAF for documents was necessary due to the absence of a valid alternative in Schema.org. The same rationale applies to the use of `foaf:OnlineAccount` for profiles.

For projects, Schema.org was considered a possible solution since `schema:Project` and `foaf:Project` are largely overlapping. However, the decision to opt for FOAF is attributable to the desire to include widely adopted ontologies in the semantic web community, external to the Schema.org sphere. Schema.org is prevalent not for the completeness or sophistication of its ontology but for enhancing the likelihood of search engines like Google Search to detect structured data on the web. Given that one of the central entities, documents, could not be described using a Schema.org class, and that FOAF provided a satisfactory alternative, FOAF was chosen for modelling projects. Additionally, FOAF is one of the most well-known and widely used ontologies in the semantic web community and supports the modelling of web-publishable documents.

For the same reasons, the choice to reuse FOAF for modelling `foaf:Person`, `foaf:Organization`, and `foaf:Agent` is justified. Furthermore, there is a constraint related to another reused ontology in Triple, PRO, which utilises these entities in role modelling.

4.1.7 PRO

The Publishing Roles Ontology (PRO) is an ontology developed to address the dynamic nature of roles in scholarly publishing. According to Peroni & Shotton (Peroni & Shotton, 2012), it is designed to describe the roles of individuals, organisations, and software agents involved in the publication process, accounting for changes in these roles over time and within specific contexts. PRO integrates with the Semantic Publishing and Referencing (SPAR) ontologies and employs the time-indexed value in context (TVC) pattern. This pattern allows PRO to model scenarios where an

entity holds a particular role during a specific time period and within a particular context, ensuring comprehensive and precise metadata representation in the Linked Data framework.

In the GoTriple ontology, the PRO ontology was used to model roles for `triple:Document`, `foaf:Project`, and `triple:Profile`.

4.1.8 Datacite

According to “DataCite-A global registration agency for research data” (Jan, 2009), DataCite is an international consortium established in late 2009 to address the need for persistent access, identification, sharing, and re-use of digital research data. The consortium's primary goals are to:

1. Establish easier access to scientific research data on the Internet: By creating a standardised approach to metadata, DataCite ensures that research data is easily accessible online.
2. Increase acceptance of research data as legitimate, citable contributions: DataCite promotes the recognition of datasets as valuable scientific outputs, encouraging their citation in scholarly work.
3. Support data archiving: DataCite aids in data archiving processes, which allow research results to be verified and repurposed for future studies.

Key to DataCite's mission is the use of persistent identifiers, particularly Digital Object Identifiers (DOIs), which provide a long-term link between a character string and a resource (such as a file, part of a file, or an abstract). This system ensures the stability and retrievability of research data across various disciplines.

The DataCite Metadata Schema is designed to facilitate accurate and consistent identification of resources for citation and retrieval, making it a cornerstone of the consortium's efforts to enhance data sharing and reuse in the academic community.

The DataCite Ontology is an ontology written in OWL 2 DL that facilitates the description of the metadata properties of the DataCite Metadata Schema Specification in RDF (Shotton et al., 2018). This enables the accurate and consistent identification of a resource for citation and retrieval purposes. The current version of the DataCite Ontology has been thoroughly revised and significantly expanded to accurately map the metadata properties from the DataCite Metadata Kernel Specification version 3.1 (DataCite Metadata Working Group, 2014) to RDF, ensuring precise metadata representation and interoperability within the Semantic Publishing and Referencing (SPAR) ontologies framework.

In GoTriple, DataCite is reused to model the identifiers of any entity within the ontology. For controlled vocabularies, DataCite is reused to define the codes corresponding to `skos:Concept` entities.

4.1.9 Schema.org

According to Barker & Campbell (Barker & Campbell, 2014) and to the official documentation (Schema.Org, 2014), Schema.org (<https://schema.org/>) is a collaborative, community-driven project that offers a collection of shared vocabularies and schemas for structured data markup on web pages. Launched in 2011 by major companies such as Google, Microsoft, Yahoo, and later Yandex, Schema.org is designed to create a unified standard for semantic markup. This vocabulary includes a wide array of item types and properties, allowing webmasters to mark up various types of content like products, events, reviews, and organisations. By incorporating Schema.org markup into HTML, webmasters enhance their content's accessibility and understandability for search engines, thus improving web data discoverability and usability.

Schema.org serves as a foundational reference for the development of the Triple ontology. This choice is not backed by specific justifications other than being a prerequisite for defining the ontology. As highlighted in the metadata analysis, the initial development of Triple's ontology is clearly oriented towards Schema.org. The reasons behind choosing Schema.org as a prerequisite can be summarised in three points:

1. *Prevalence*: Schema.org is one of the most widespread and recognized standards. Many developers, even those not particularly skilled in ontological modelling, recognize Schema.org as essential for enhancing the indexing of their resources on the web. Thus, Schema is valued more for its capabilities in communicating with Google's search engine than for the modelling it proposes.
2. *Coverage*: The schema covers the most common and demanded modelling cases, so many developers do not even need to look for standards closer to modelling cases that deviate from those presented by Schema.org.
3. *Readability*: The documentation of Schema.org offers unquestionably better readability and usability compared to other resources, including those produced by W3C. Additionally, it is possible to search for terms of interest using a search box on the website itself. Compared to other documentation, this presents strong incentives for reuse.

In Triple, Schema.org is reused, when possible, to model nearly all scenarios presented in the following chapter. When Schema.org presented restrictions or insufficient modelling, other equally widespread and recognized ontologies in the field of the semantic web were considered.

4.1.10 TVC

The Time Interval ontology pattern (TI) (Peroni et al. 2012) is a Content Ontology Design Pattern (CP) that represents time intervals. It allows for answering questions related to temporal extent, such as “What is the starting date of the interval?” and “What is the ending time of the interval?”. TI can be composed with other Content OPs when temporal aspects need to be represented. Its elements are:

- `ti:TimeInterval`. Any region in a dimensional space that represents time.
- `ti:hasIntervalDate`. A datatype property that encodes values from `xsd:date` for a time interval. The same time interval can have more than one `xsd:date` value (e.g., begin date, end date, date at which the interval holds), as well as dates expressed in different formats (`xsd:gYear`, `xsd:dateTime`, etc.).
- `ti:hasIntervalStartDate`. The start date of a time interval.
- `ti:hasTimeEndInterval`. The end date of a time interval.

In Triple, the Temporal Interval (TI) was used to accurately describe the temporal placement of projects. Additionally, it is one of the Ontology Design Patterns (ODP) reused from the PRO ontology.

4.1.11 Textual Data

In Triple, one of the most important elements to model is the textual elements. Considering the reused ontologies, textual elements are represented by three types of datatypes:

- `rdfs:Literal`
- `schema:Text`
- `xsd:string`

For modelling purposes, the decision was made to simplify the chosen datatypes, opting for `rdfs:Literal`. The `schema:Text` term from Schema.org is designed for web data

interchange and provides a straightforward way to represent text in a context easily understandable by web search engines. However, it lacks the formal semantic rigour found in other vocabularies. The `xsd:string`, part of the XML Schema Definition (XSD) language, offers a precise definition of a character sequence, ensuring compatibility and validation across XML-based systems. Nevertheless, its use is largely limited to syntactic validation rather than semantic interpretation.

On the other hand, `rdfs:Literal`, a fundamental construct of RDF Schema (RDFS), not only accommodates plain text but also includes datatypes and language tags, enhancing the expressiveness and granularity of the data. Considering that GoTriple gathers a vast amount of data, which despite the normalisation processes highlighted in Deliverable 2.5 Data Enrichment (De Santis, 2022), requires the modelling of an ontology with datatypes that can accommodate the widest range of cases. Compared to other datatypes, `rdfs:Literal` allows the right degree of flexibility and compatibility with semantic web standards.

4.2 Application Scenario

In the following sections, several application scenarios of the Triple ontology, described in its essential components in the previous sections, will be illustrated. Each section will be structured as follows:

- *Scenario*: A small scenario will be illustrated, involving the entities and relationships modelled by the Triple ontology.
- *RDF Turtle*: The scenario will be presented in RDF Turtle format ([https://en.wikipedia.org/wiki/Turtle_\(syntax\)](https://en.wikipedia.org/wiki/Turtle_(syntax))).
- *Visualisation*: The scenario defined in the previous point will be visually illustrated using Graffoo (Graffoo, <http://www.essepuntato.it/graffoo>).
- *SPARQL Query*: A use case and the corresponding SPARQL query will be illustrated.

4.2.1 Relations between Document expressions and manifestations

To illustrate the following example, let's consider an academic article (`triple:Document`). This article has two manifestations (`fabio:Manifestations`): a PDF version and a printed version. These manifestations are grouped together within a cluster (`triple:Cluster`).

```
ex:document_expression_1 a triple:Document ;  
    frbr:embodiment ex:document_manifestation_1 ;  
    frbr:embodiment ex:document_manifestation_2 .  
  
ex:document_manifestation_1 a fabio:Manifestation ;  
    collectionentity:isMemberOf ex:cluster_1 ;  
  
ex:document_manifestation_2 a fabio:Manifestation ;  
    collectionentity:isMemberOf ex:cluster_1 ;  
  
ex:cluster_1 a triple:Cluster
```

Listing 2: RDF application scenario 1.

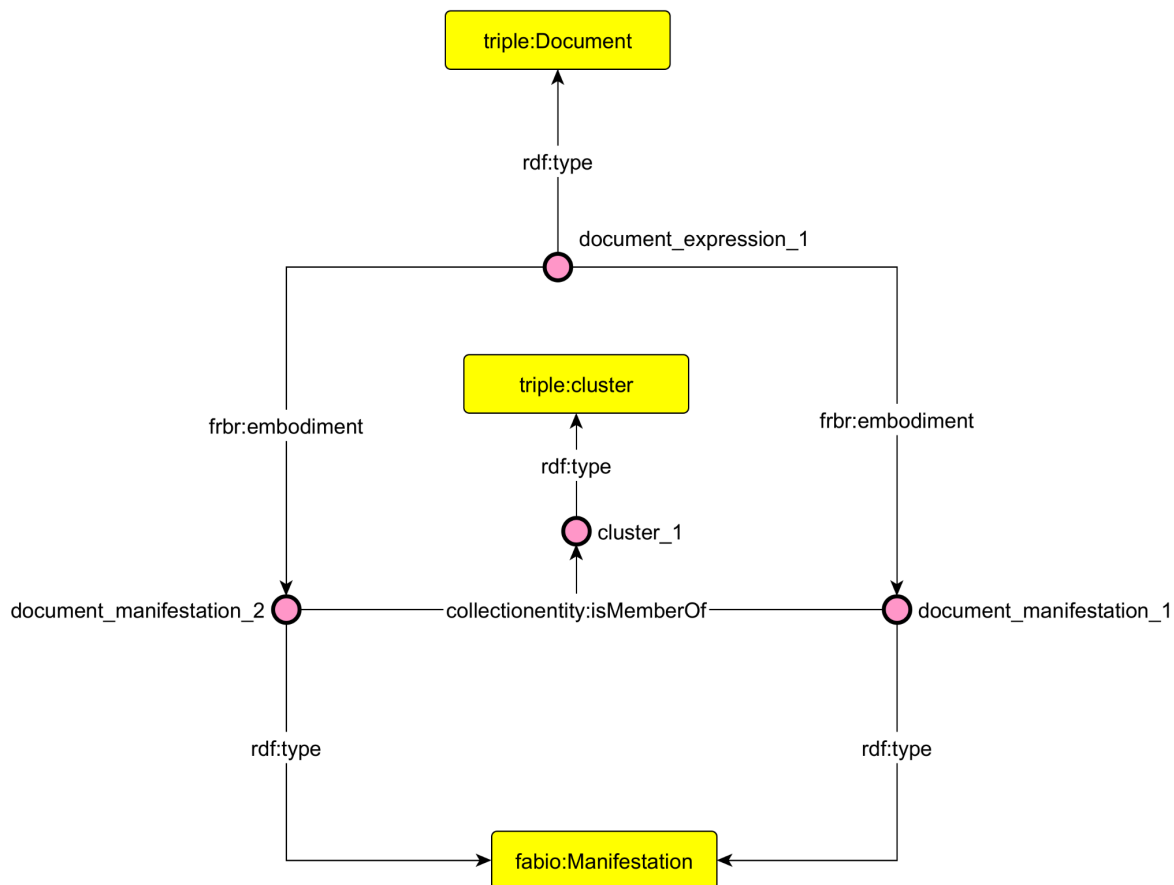


Figure 1: application scenario 1.

An interesting use case involves a user who finds an academic article of interest on the GoTriple platform. At this point, the user wishes to understand which manifestations of the document are

stored in the GoTriple database. The user can then formulate the following query to retrieve the desired results.

```
SELECT ?expression ?manifestation ?cluster
WHERE {
    ?expression a triple:Document ;
                frbr:embodiment ?manifestation .
    ?manifestation a fabio:Manifestation ;
                collectionentity:isMemberOf ?cluster .
    ?cluster a triple:Cluster .
}
```

Listing 3: SPARQL query application scenario 1.

4.2.2 Relations Between Documents, Roles, Profiles, Online Accounts, and Agents

To illustrate the following example, we will partially revisit the previous example. The only element omitted is the clusters, which are unnecessary for this illustration. Let us revisit the previous academic article (`triple:Document`). This document corresponds to two manifestations: one in PDF format and the other in print format. This document has an associated person (`foaf:Person`) indicated as the author (`pro:Role`). On GoTriple, this author is associated with an account (`foaf:OnlineAccount`), created after the user registered on the platform and claimed their publications. However, the author's name is written differently in the various manifestations of the documents (the PDF and print versions) of which they are the author.

```
ex:document_expression_1 a triple:Document ;
    frbr:embodiment ex:document_manifestation_1 ;
    frbr:embodiment ex:document_manifestation_2 ;
    pro:isDocumentContextFor role_in_time_1 .

ex:role_in_time_1 a pro:RoleInTime ;
    pro:isHeldBy ex:agent_1 ;
    pro:withRole pro:author .

pro:author a pro:Role .

ex:document_manifestation_1 a fabio:Manifestation ;
    pro:isDocumentContextFor profile_1 .

ex:profile_1 a triple:Profile ;
    pro:isHeldBy ex:agent_1 ;
    pro:withRole pro:author ;
    triple:name ex:profile_name_1 .

ex:document_manifestation_2 a fabio:Manifestation ;
    pro:isDocumentContextFor profile_2 ;

ex:profile_2 a triple:Profile ;
    pro:isHeldBy ex:agent_1 ;
    pro:withRole pro:author ;
    triple:name ex:profile_name_2 .

ex:agent_1 a foaf:Person ;
    foaf:account ex:account_online_1 .

ex:account_online_1 a foaf:OnlineAccount .

ex:profile_name_1 a triple:profileName .

ex:profile_name_2 a triple:profileName .
```

Listing 4: RDF application scenario 2.

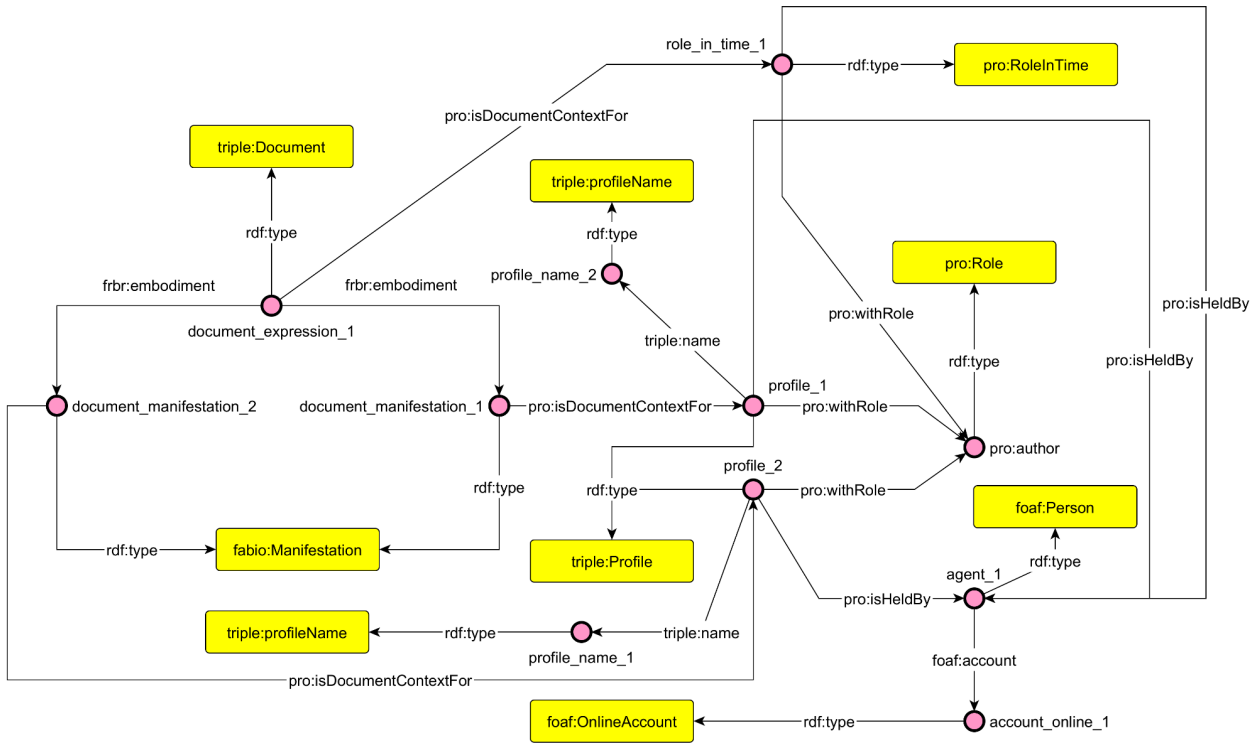


Figure 2: application scenario 2.

Consider this use case: an imagined user is interested in retrieving the various declinations of an author's name given one of their publications. Suppose the user, who is trying to obtain this information, is interested in creating their own system for author name disambiguation related to a set of articles on which they need to conduct a quantitative analysis. By formulating the following SPARQL query, the user can retrieve all the variants of the author's name that GoTriple encountered during the data ingestion phase.

```
SELECT ?document_expression ?profile_name
WHERE {
  ?document_expression a triple:Document ;
                      frbr:embodiment ?document_manifestation .
  ?document_manifestation a fabio:Manifestation ;
                      pro:isDocumentContextFor ?profile .
  ?profile a triple:Profile ;
          pro:isHeldBy ?agent ;
          triple:name ?profile_name .
  ?agent a foaf:Person .
}
ORDER BY ?document_expression
```

Listing 5: SPARQL query application scenario 2.

4.2.3 Connecting Documents to Controlled Vocabularies

This example explores the connection between `triple:Documents` and controlled vocabularies, focusing on the vocabulary of academic disciplines.

Consider an academic article. This article can be connected to an academic discipline. This discipline is described through the connection to external vocabularies that, due to their comprehensiveness and completeness, possess greater authority.

```
ex:document_expression_1 a triple:Document ;
    sioc:topic ex:concept_1 .

ex:concept_1 a skos:Concept ;
    skos:inSchema triple:discipline_scheme ;
    skos:exactMatch ex:wikidata_concept_1 ;
    skos:closeMatch ex:wikidata_concept_2 .

ex:wikidata_concept_1 a skos:Concept .
ex:wikidata_concept_2 a skos:Concept .
ex:discipline_scheme a skos:ConceptScheme .
```

Listing 6: RDF application scenario 3.

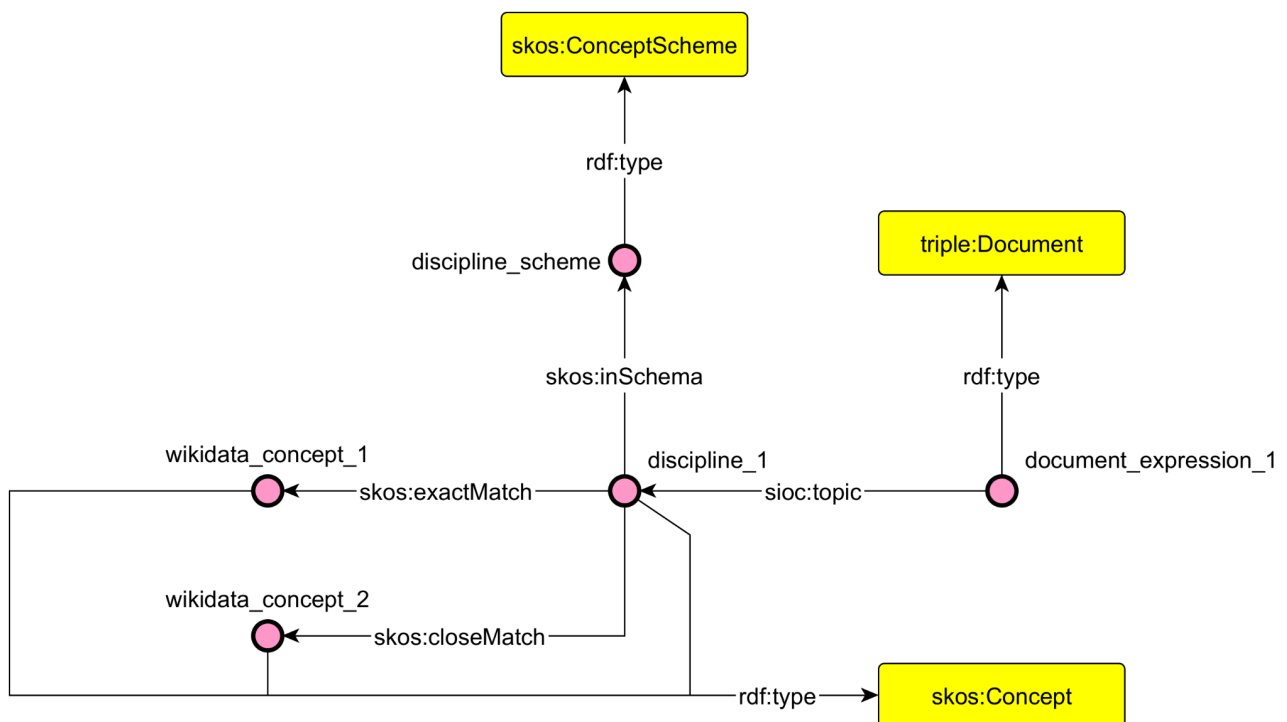


Figure 3: application scenario 3.

Given this example, a user might be interested not only in investigating which discipline the academic article is associated with but also in extending their research to the external vocabularies

that describe this discipline. Specifically, in the example provided, Wikidata is used. From Wikidata, a user could extend their research to the corresponding Wikipedia page.

```
SELECT ?concept ?exactMatch ?closeMatch

WHERE {

    ?concept skos:inScheme ex:discipline_scheme .

    OPTIONAL { ?concept skos:exactMatch ?exactMatch }

    OPTIONAL { ?concept skos:closeMatch ?closeMatch }

}
```

Listing 7: SPARQL query application scenario 3.

Chapter 5: Development

In this section, we delve into the various iterations developed with the domain expert, applying the SAMOD (SAMOD, <https://essepuntato.it/samod/>) methodology outlined in Chapter 3. Each subsection is structured as follows:

1. *Motivating Scenario*: This part defines the scenario to be represented ontologically. It does not delve into specific ontological choices but outlines the requirements to be considered during modelling.
2. *Competency Questions*: This section lists questions that the model must be able to answer through SPARQL queries.
3. *Description*: Here, we detail the ontological choices made, specifying which classes and properties were used to represent the motivating scenario. A graphical visualisation of this ontology section, created using the Graffoo tool (Graffoo, <http://www.essepuntato.it/graffoo>) (Falco et al., 2014), will support this description.

5.1 Scenario 1: Documents

This scenario is devoted to the definition of one of the fundamental entities of the data model, the document. This iteration allows you to model a document with some basic metadata.

5.1.1 Motivating Scenario

In the GoTriple environment, the entity document refers to a broad spectrum of records obtained from data providers. These records encompass a wide variety of scholarly and research-oriented materials. The technical description of a document within GoTriple encompasses several key aspects:

- *Languages*: The language attribute of a document is critical for its categorization and accessibility. GoTriple incorporates a controlled vocabulary that covers a wide range of languages, ensuring accurate representation and identification. This includes primary languages like Croatian, English, French, German, Greek, Italian, Polish, Portuguese, Slovenian, Spanish, and Ukrainian, along with other common languages such as Arabic, Dutch, Swedish, and more. Special labels like "other" and "undefined" are also used to categorise languages not specifically listed or when language data is missing. Languages are

formatted in ISO-639-1 two-character codes, aligning with the platform's language recognition and translation services.

- *Identifiers*: Each document in GoTriple is associated with several identifiers, enhancing its traceability and accessibility. These identifiers include:
 - *Local Identifier*: A unique identifier assigned within the GoTriple system, serving as a primary reference point for the document within the platform.
 - *DOI* (Digital Object Identifier): A widely recognized identifier that provides a persistent link to the document's location on the internet. DOIs are crucial for ensuring long-term accessibility and are commonly used in academic and research settings.
 - *Full Document URL*: The direct URL to the document itself, providing immediate access to the document for viewing or download.
 - *Source URL*: This URL points to the original source of the document, offering a link to where the document was initially published or hosted.
 - *Landing Page URL*: The URL of the landing page that provides descriptive information about the document, often including metadata, abstracts, and links to the full document or related resources.
- *Title and Abstract*: Each document in GoTriple is associated with a title and an abstract.
- *Date published*: The publication date indicates the date on which the document was published.

5.1.2 Competency questions

- What are the identifiers of a specific document
- What language is the document written in?
- What is the abstract and title of the document?
- When was the document published?

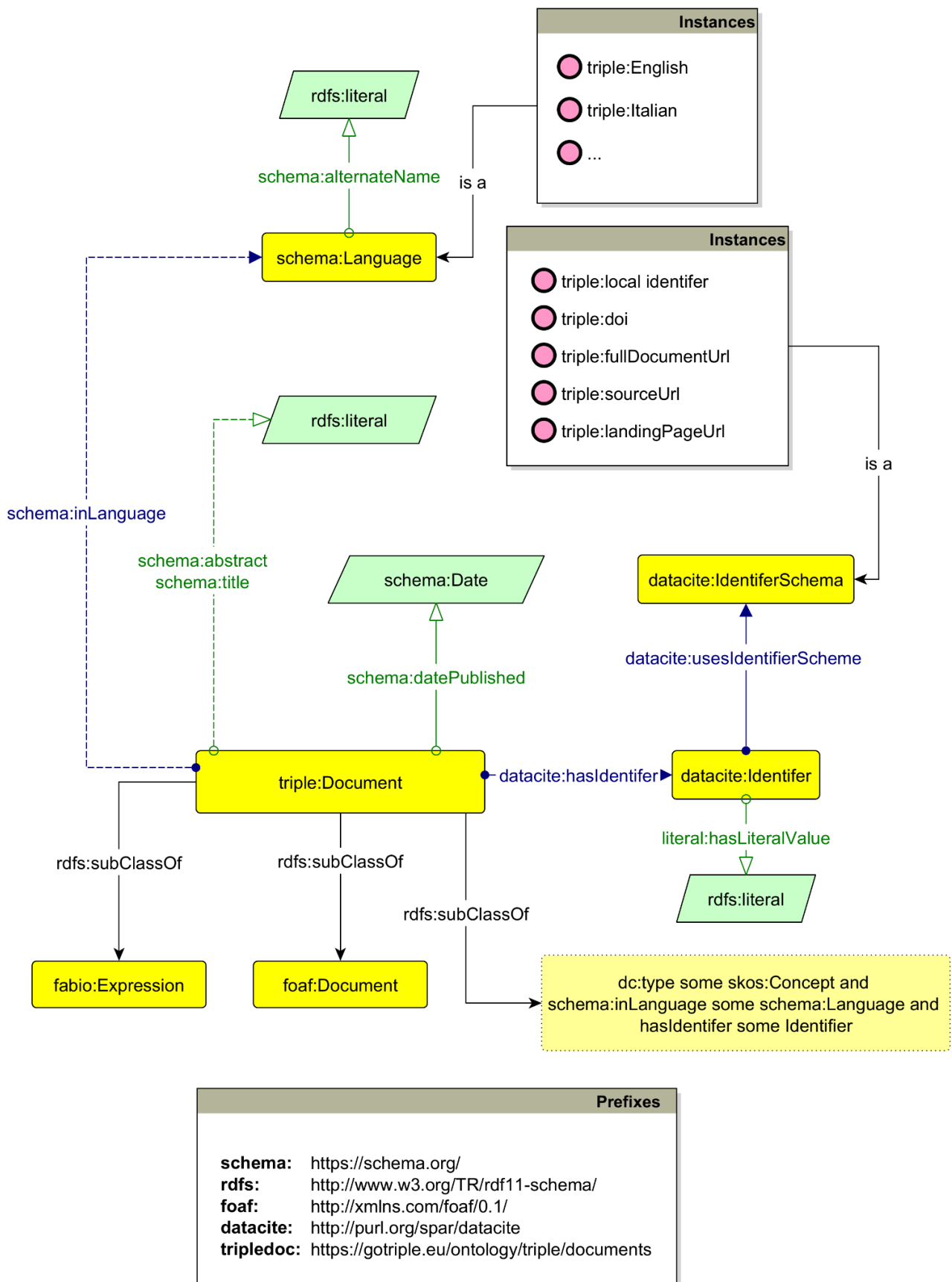


Figure 4: iteration 1.

5.1.3 Description

The materials for this iteration can be found in the directory located at <https://github.com/AlessandroBertozzi/TRIPLE-ontology/tree/main/development/01>. The entities depicted in the Graffoo image (Figure 4) represent the fundamental entities and relations of this iteration.

The central role of iteration is represented by the document. As extensively described in the previous chapter (see Chapter 4.1.3), the `triple:Document` is represented both at the level of expression (`fabio:Expression`) and manifestation (`fabio:Manifestation`).

For the representation of titles and abstracts of a `triple:Document`, we employed the Schema.org vocabulary and the `rdfs:Literal` datatype. The title is represented using the `schema:title` property, ensuring a clear and standardised identification of the document's main title. Similarly, the abstract, which provides a summary of the document's content, is represented using the `schema:abstract` property. Both the title and abstract values are connected to the `rdfs:Literal` datatype to maintain consistency across the data model.

Each document is written in a specific language. For representation purposes, we have chosen to reuse the Schema.org design pattern, specifically `schema:Language`. The entity `triple:Document` is connected via the property `schema:inLanguage`. The `schema:Language` entity is linked to a formal language code expressed in BCP 47 (https://it.wikipedia.org/wiki/Codice_di_lingua_IETF) through the data property `schema:alternateName`. This property is connected to an `rdfs:Literal` datatype. Although Schema.org typically uses `schema:Text` for such representations, we opted to use `rdfs:Literal` for all textual data types to maintain model consistency. A consideration in this design is the placement of language at the expression level rather than at the manifestation level. The manifestation level is where the document takes on a physical form through a specific format and is expressed in a particular language. In this model, the language associated with an expression of the work represents the language in which the original expression of the work was conceived (e.g., the Lombard dialect for "The Betrothed" by Alessandro Manzoni). At the manifestation level, this expression of the work can be rendered in a different language (e.g., the English edition of "The Betrothed"). For this data model, considering the manifestation level was deemed unnecessary.

For each `triple:Document`, it is possible to specify the publication date. To represent the temporal placement of the document, the entity `foaf:Document` has been connected using the `schema:datePublished` design pattern defined in schema.org. The design pattern

`schema:datePublished` is a specific property in the `schema.org` vocabulary used to denote the publication date of a `schema:CreativeWork`, such as a document. This property expects a value of the data type `schema>Date`, which conforms to the ISO 8601 date format (YYYY-MM-DD).

As illustrated in the motivating scenario, each document can be associated with various identifiers. To connect an identifier to the `triple:Document` entity, we reused the pattern from the DataCite Ontology. This involves two classes: `datacite:Identifier` and `datacite:IdentifierSchema`. The various types of identifiers identified in the motivating scenario are included as individuals within the `datacite:IdentifierSchema` class. The value of the identifier is represented by the datatype `rdfs:Literal`, which is connected to `datacite:Identifier` through the property `literal:hasLiteralValue`. Therefore, a document is connected to one or more identifiers, and each identifier is connected to a schema that represents its type.

5.2 Scenario 2: Controlled Vocabulary

In this iteration, we illustrate how the management of controlled vocabularies has been modelled. By the end of this iteration, the model allows for the description of all controlled vocabularies used to describe a document. This approach ensures that each vocabulary, essential for standardising terminologies and concepts across documents, is accurately and consistently represented. The controlled vocabularies are crucial for improving the reusability and interoperability of documents within the GoTriple platform.

5.2.1 Motivating Scenario

Every document can be associated with one or more of the following entities: licence, access conditions, type of document, and field of discipline. The data sources from which individual documents are retrieved contain an extremely variable number of these entities. To reduce the complexity and variability of this information, the approach adopted was to extract the most frequent terms and create controlled vocabularies. Therefore, for each of the entities listed above, a controlled vocabulary has been defined:

1. *Licence*: This refers to the legal permissions and restrictions associated with a document. A licence dictates how the document can be used, shared, modified, and distributed. For instance, some documents might be under open licences allowing free use and distribution, while others could have more restrictive licences that limit their use to certain conditions or require payment or attribution.
2. *Access Conditions*: These are the terms that govern how and by whom a document can be accessed. Access conditions might include restrictions based on user credentials, subscription status, geographical location, or other criteria. For example, some documents might be publicly accessible, while others could be available only to certain academic institutions or paid subscribers.
3. *Type of Document*: This refers to the category of the document. It includes distinctions such as research papers, articles, technical reports, books, or datasets. Each type of document has its own structure, purpose, and audience. For instance, a research paper might present new findings in a specific field, whereas a technical report might provide detailed technical information about a project or study.
4. *Field of Discipline*: This entity relates to the academic or professional field to which a document belongs. It denotes the subject area or the specialised domain of knowledge

covered by the document. Examples of disciplines include economics, history, and many others. This categorization is crucial for researchers and professionals seeking information specific to their field of study or interest.

In order to define the common basic structure of each controlled vocabulary, the following structure is required within this scenario. Each term listed in the vocabulary must have the following capabilities:

- *Connection to a Vocabulary*: Each term must be associated with a specific controlled vocabulary. This ensures that it is categorised correctly and is part of a structured collection of standardised terms.
- *Unique Identifier*: Every term in the vocabulary should have a unique identifier.
- *Ability to Connect to One or More External Entities*: Each term should have the capacity to link to external entities in two specific ways:
 - *Close Match*: This implies that the term in the vocabulary is nearly equivalent to, but not exactly the same as, a term in an external system or entity. A close match suggests a high degree of similarity or relevance, though the terms might not be identical.
 - *Exact Match*: This indicates that the term in the controlled vocabulary is precisely the same as a term in an external entity. An exact match is used when the terms are interchangeable, with no variation in meaning or context.

5.2.2 Competency Questions

- What types of documents are most frequently found in the dataset?
- To which academic or professional field does a given document belong?
- Which documents are related to the field of history?
- What are the access conditions for a given document?
- Which documents are categorised as research papers?

Prefixes	
schema:	https://schema.org/
foaf:	http://xmlns.com/foaf/0.1/
datacite:	http://purl.org/spar/datacite
skos:	http://www.w3.org/2004/02/skos/core#
triple:	https://gotriple.eu/ontology/triple
litre:	http://www.essepuntato.it/2010/06/literalreification/

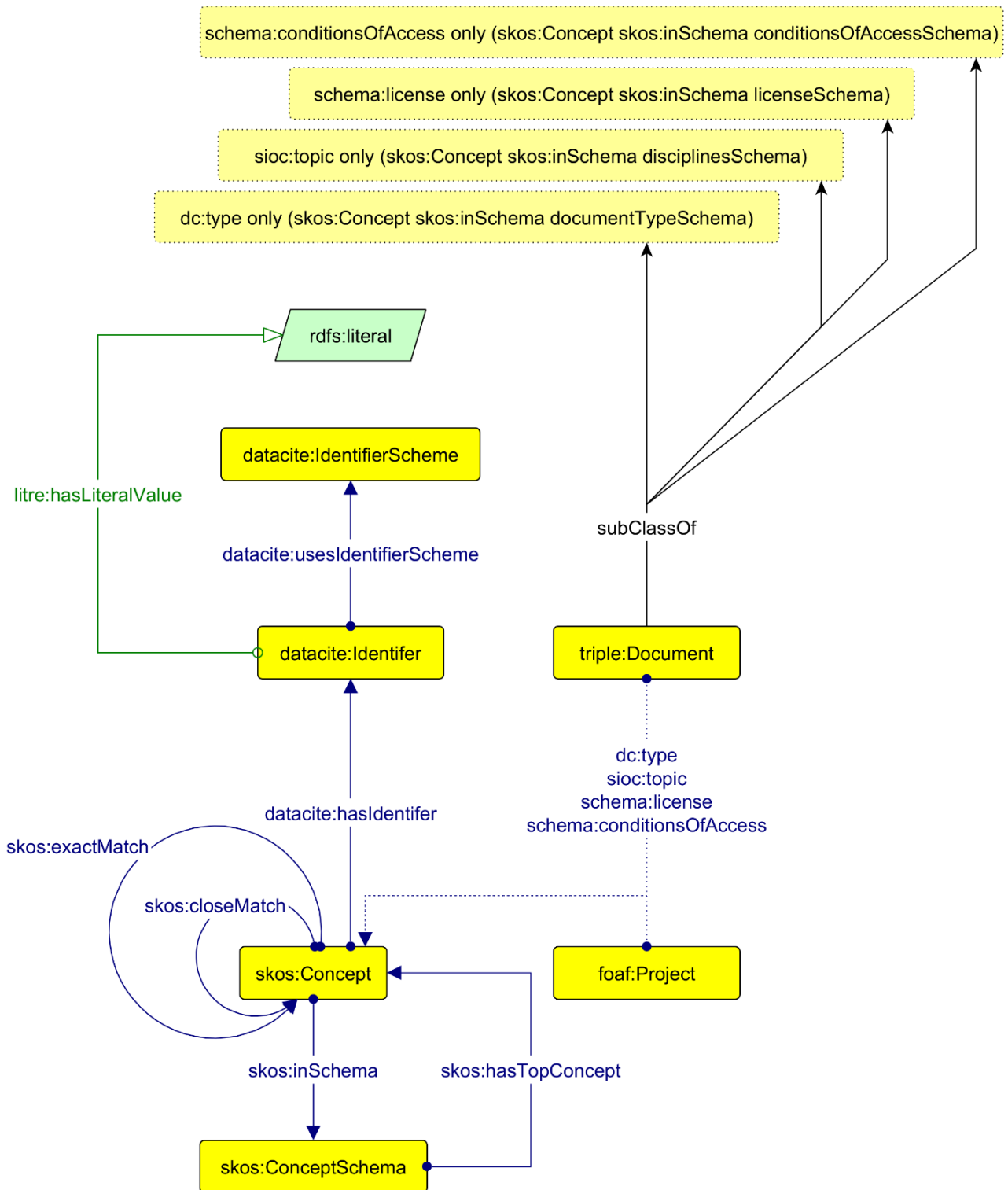


Figure 5: iteration 2.

5.2.3 Description

The materials for this iteration can be found in the directory located at <https://github.com/AlessandroBertozzi/TRIPLE-ontology/tree/main/development/02>. The entities depicted in the Graffoo image (Figure 5) represent the fundamental entities and relations of this iteration.

Each vocabulary is formalised as an individual under the `skos:ConceptScheme` class. In total, four vocabularies are defined: `triple:license`, `triple:discipline`, `triple:conditionsOfAccess`, and `triple:documentTypes`. A vocabulary consists of a list of terms. Each term is defined as an individual under the `skos:Concept` class and is linked to a vocabulary via the `skos:inScheme` relationship.

A key strength of SKOS is the ability to connect its concepts to other concepts in different controlled vocabularies. The chosen relationships to represent these connections are `skos:exactMatch`, for linking concepts whose definitions make them interchangeable, and `skos:closeMatch`, for linking concepts that are not interchangeable but similar.

Each concept can be described using annotations. Specifically, the annotation properties `rdfs:label` and `skos:definition` are reused. Both can specify the language in which they are written, allowing the vocabulary to present a `skos:Concept` in multiple languages.

Additionally, each term can specify a unique code that identifies it. To model this scenario, the DataCite Ontology was integrated with SKOS. The same pattern used for modelling document identifiers is employed: `datacite:Identifier` connected via the object property `datacite:usesIdentifierScheme` to the class `datacite:IdentifierScheme`.

Controlled vocabularies are used to describe both the `triple:Document` class and the `foaf:Project` class. For `foaf:Project`, it is possible to specify the disciplinary scope via the object property `sioc:topic` and the concepts related to the `triple:discipline` vocabulary. The same object property links `triple:Document` to the concepts in the controlled vocabulary `triple:discipline`. Additionally, `triple:Document` is connected via the object properties `dc:type`, `schema:license`, and `schema:conditionsOfAccess` to the concepts in the vocabularies document types, license, and conditions of access, respectively.

5.3 Scenario 3: Roles

In this iteration, document-related roles and agents are presented. At the end of this iteration, the ontology is able to correctly represent the agents and with what roles they are involved in the publication and dissemination cycle of a `triple:Document`.

5.3.1 Motivating Scenario

The life cycle and creation of a document in GoTriple involve various roles, each playing a significant part in its development, dissemination, and maintenance. These roles, which can be assumed by either organisations or individuals, are fluid and may change over time. They include:

1. *Author*: The individual or group primarily responsible for creating the content of the document. Authors are the intellectual source of the research and ideas presented in the document.
2. *Contributor*: These are individuals or entities that have played a significant, but not primary, role in the creation of the document. Their contributions can be in various forms, such as providing data, writing assistance, or other forms of support.
3. *Publisher*: The publisher is responsible for the distribution and dissemination of the document. This role often involves tasks such as editing, design, marketing, and ensuring the document reaches its intended audience.
4. *Provider*: This refers to the entity that makes the document available within the GoTriple platform. Providers may be different from the original publishers, especially in cases where documents are sourced from third-party databases or repositories.
5. *Aggregator*: An aggregator in the context of GoTriple collates various documents from different sources. This role is crucial for enhancing the platform's database by bringing together diverse SSH resources from multiple origins.
6. *Primary Producer*: This is the original source or creator of the document, often before it undergoes processing or publishing. The primary producer is usually responsible for the initial creation and compilation of the content.
7. *Funder*: Funders are individuals or organisations that provide financial support for the creation, research, or publication of a project. Their role is vital in enabling the research and dissemination process, especially in academic and scholarly contexts.

Each of these roles contributes to the lifecycle of a document in GoTriple, from its initial creation to its eventual dissemination and use. This dynamic ecosystem ensures that documents are not only rich in content but also supported by a network of contributors and facilitators, enhancing the overall value and accessibility of SSH resources on the GoTriple platform.

In a technical framework like GoTriple, the role system for document creation and lifecycle management is defined by three key parameters: the temporal collocation, the associated entities (persons or organisations), and the specific functions or responsibilities of each role.

1. *Temporal Collocation:*

- a. Each role is aligned with a specific phase in the document's lifecycle. This temporal placement defines when a particular role is active and influential in the process of document creation, modification, dissemination, or preservation.
- b. The lifecycle phases typically include the initial creation or conceptualization phase, the development or production phase, the publication or release phase, and the archival or preservation phase.

2. *Associated Entities:*

- a. Roles can be assumed by individuals, such as researchers or contributors, or by organisations, including academic institutions, publishing houses, data repositories.
- b. The entity associated with a role may vary based on the nature of the document, the field of study, and the resources available for the project.

3. *Role Specification:* Each role comes with a set of specific functions or responsibilities that contribute to the document's development.

In this role system, the interplay between the temporal collocation, the entities involved, and their specific roles ensures a dynamic yet structured progression of a document from its inception to its ultimate utilisation and archiving. This systematic approach is crucial in managing the complex processes involved in the handling of scholarly and research documents, particularly in a multifaceted platform like GoTriple.

5.3.2 Competency Questions

- Who is the publisher of the article?
- What roles are associated with the article?
- When was the author of the article authored? Did he or she have a change of role?

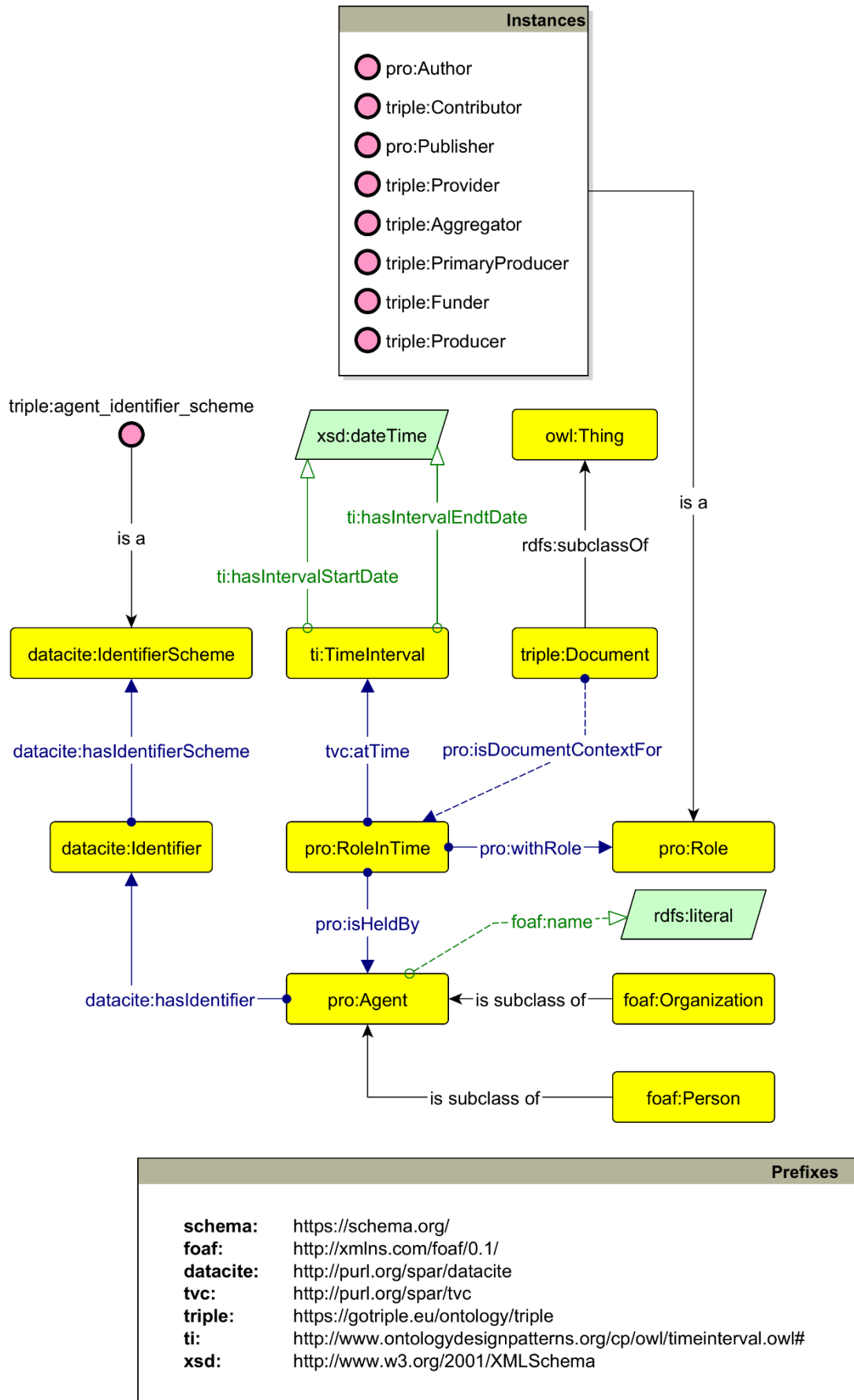


Figure 6: iteration 3.

5.3.3 Description

The materials for this iteration can be found in the directory located at <https://github.com/AlessandroBertozzi/TRIPLE-ontology/tree/main/development/03>. The entities depicted in the Graffoo image (Figure 6) represent the fundamental entities and relations of this iteration.

To describe the scenario illustrated in this iteration, we decide to reuse an existing ontology: PRO. The Publishing Roles Ontology (PRO), as mentioned in Chapter 3, characterises the roles of agents—people, corporate bodies, and computational agents—in the publication process. It specifies an agent's role (e.g., author, editor) concerning a bibliographic entity or institution and the time period during which the role is held.

The document connects to statements defined by PRO through the property `pro:isDocumentContextFor`, linking a document to the `pro:RoleInTime` class. The `pro:RoleInTime` must specify a role, a time range, and an agent. The ontology specifies 31 roles under the `pro:Role` class. Of these, 2 roles are reused: `pro:Author` and `pro:Publisher`. Additionally, 6 roles are added as individuals to the `pro:Role` class, defined within the Triple namespace: `triple:producer`, `triple:Founder`, `triple:PrimaryProducer`, `triple:Aggregator`, `triple:Provider`, and `triple:Contributor`.

For specifying the time range, we use the `ti:TimeInterval` class from the TI ontology. Finally, for representing the agent associated with a `pro:RoleInTime`, we use the `pro:Agent` class. An agent can be a person or, in certain cases, an organisation. To handle these `pro:Agent` types, two subclasses of `pro:Agent` have been created: `foaf:Person` and `foaf:Organization`.

5.4 Scenario 4: Subjects Coverage

In this iteration, the metadata that allows defining the subjects covered by a `triple:Document` are presented. At the end of this iteration, the ontology is able to correctly represent which temporal, spatial, and keyword topics describe a document.

5.4.1 Motivating Scenario

In the GoTriple platform, documents must have metadata associated with the subjects discussed in the article content. This need arises for two reasons: the data retrieved from the data providers had these entities. To facilitate the search of documents through metadata. Having metadata, and thus entities, that manage the subjects covered by a document are highly effective tools in search and filtering. Three types of metadata are used to describe subject coverage: temporal coverage, spatial coverage and keywords. Temporal coverage refers to the time span or historical period to which the document content refers. This can be a specific date. It allows users to understand the historical context or specific period relevant to the research or discussion presented in the document. discussion presented in the paper. This time aspect is useful for researchers focusing on historical trends, chronological analyses or specific time studies. Spatial coverage refers to the geographical focus or area of relevance of the document. This may be a country, a region, a city or even a specific location that is central to the content of the document. This geographical aspect is useful in understanding the regional relevance or geographical scope of the information or research presented. It is especially important for studies that are location-specific or when researchers are interested in SSH resources related to specific geographical areas. Keywords are words that capture the essence of the document, making the article more easily searchable and filterable through special search filters. Each document can have one or more associated keywords.

5.4.2 Competency Question

- What places are associated with the content of the article?
- What events or dates are associated with the content of the article?
- With what keywords are search artefacts described?

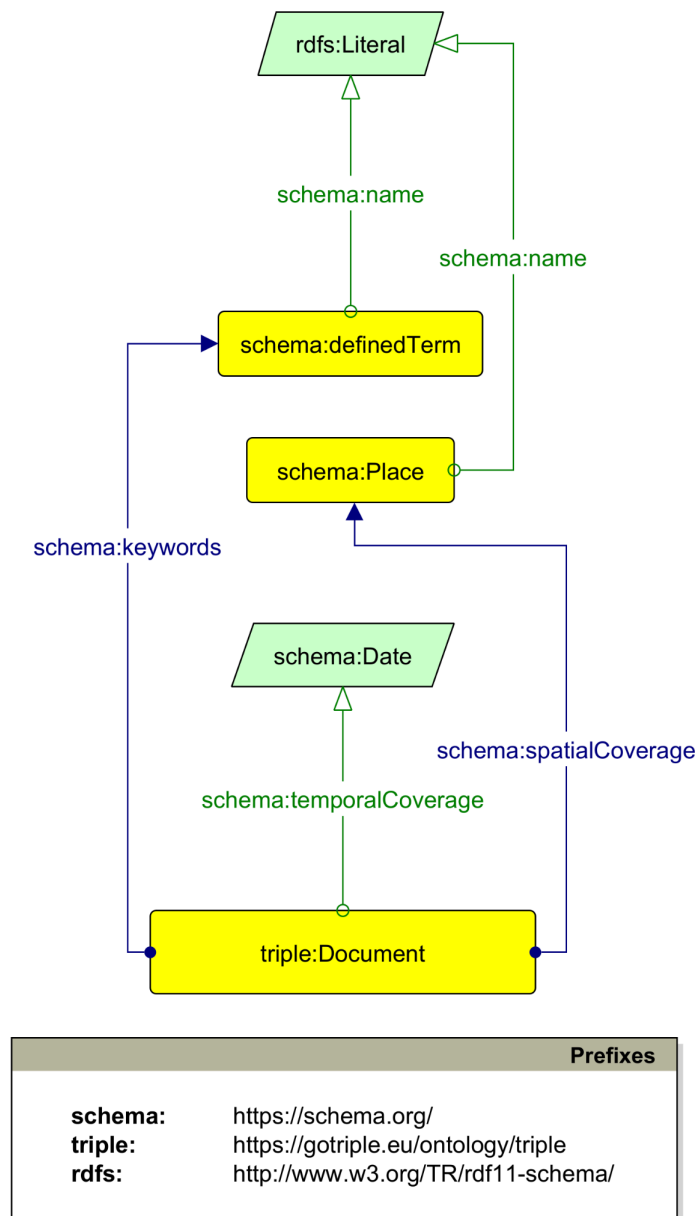


Figure 7: iteration 4.

5.4.3 Description

The materials for this iteration can be found in the directory located at <https://github.com/AlessandroBertozzi/TRIPLE-ontology/tree/main/development/04>. The entities depicted in the Graffoo image (Figure 7) represent the fundamental entities and relations of this iteration.

For modelling the scenario in this iteration, the Schema.org ontology was utilised. Additionally, Schema.org provided sufficient statements to describe the scenario effectively. The only modification made was to the expected datatype for `schema:TemporalCoverage`. The expected type is `schema:DateTime`, which is too restrictive for our use case. The `DateTime` format requires a combination of date and time of day. Therefore, we opted to change this datatype to `schema:Date`.

To describe the topic related to space, we used `schema:Place`, which allows specifying the name via the data property `schema:Text`. For representing keywords, we employed the design pattern expected by the object property `schema:keywords` from Schema.org. Finally, to specify a topic related to time, we used the data property `schema:TemporalCoverage`, where the expected type is either `schema:Date` or `schema:Text`. This allows for defining both a precise date and a simple string (e.g., WWII).

5.5 Scenario 5: Discarded Entities and Documents Clustering

Two specific elements of the Triple ontology are presented in this iteration. Discarded keywords and document clusters. This iteration makes it possible to correctly represent document clusters and what kind of relationship they define with triples:Document. It also provides a representation pattern for entities that predict certain individuals as “discarded”.

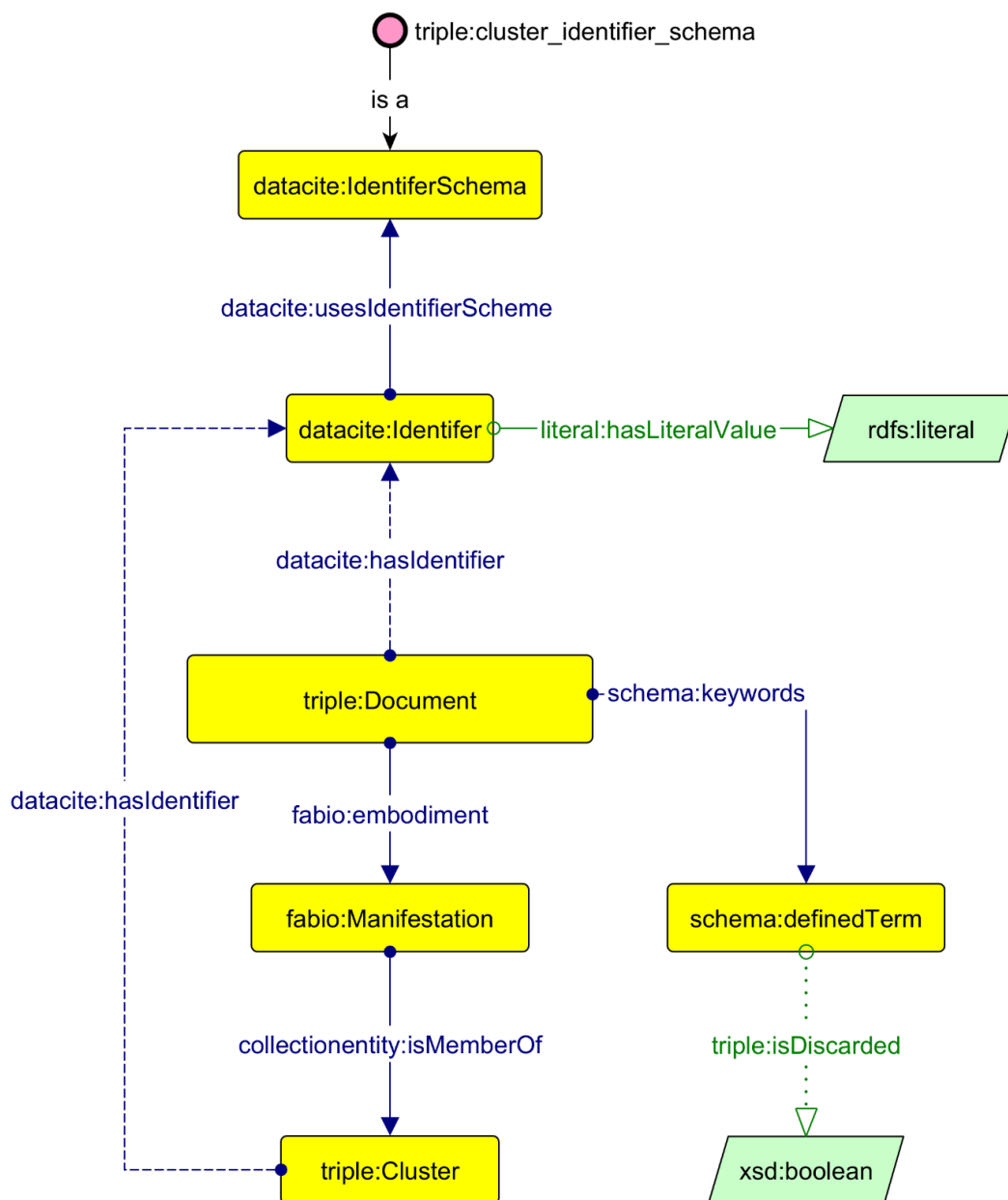
5.5.1 Motivating Scenario

One of the most frequent problems when harvesting and ingesting data from third-party data sources is the duplication of documents. Indeed, the external sources taken by the GoTriple platform to aggregate resources may contain overlapping data. One difficulty is recognising when two documents from different external sources are the same document. In the event that two or more similar documents are found to be the same, the concept of a cluster has been defined in GoTriple. The cluster is a set of documents to which an identifier corresponds, within which documents can be retrieved, occasionally with different identifiers, but traceable to the same document.

Another problem in GoTriple, again attributable to the number and diversity of external sources used, is the unusability of certain data. In particular, we refer to one entity: keywords. Some keywords were decided not to be considered as useful for research purposes within the platform. For this reason, it was decided not to make them useful for searching and filtering documents. For the sake of transparency, however, it was not decided to permanently remove them from the platform.

5.5.2 Competency Question

- How many articles does the cluster contain?
- What keywords related to the article were discarded?



Prefixes	
schema:	https://schema.org/
foaf:	http://xmlns.com/foaf/0.1/
datacite:	http://purl.org/spar/datacite
skos:	http://www.w3.org/2004/02/skos/core#
triple:	https://gotriple.eu/ontology/triple
litre:	http://www.essepuntato.it/2010/06/literalreification/
collectionentity:	http://www.ontologydesignpatterns.org/cp/owl/collectionentity.owl

Figure 8: iteration 5.

5.5.3 Description

The materials for this iteration can be found in the directory located at <https://github.com/AlessandroBertozzi/TRIPLE-ontology/tree/main/development/05>. The entities depicted in the Graffoo image (Figure 8) represent the fundamental entities and relations of this iteration.

The scenario necessitates modelling two specific entities in the Triple ontology: Cluster and Discarded.

A Cluster is a collection of `triple:Document`. More precisely, it refers to the `fabio:Manifestation` of `triple:Document`. The ingestion pipelines gather data from external providers. The ingestion and normalisation pipeline is responsible for identifying and grouping documents that can be considered duplicates. Whether a document is considered a duplicate depends on a series of metadata, which excludes the format. Excluding the format, necessary for distinguishing the `fabio:Manifestation` of `triple:Document`, thus transforms clusters into a container of `triple:Document`. This container will effectively include both duplicates and manifestations. Given these considerations, the Triple ontology defines a class `triple:Cluster` linked to the `fabio:Manifestation` of `triple:Document` via the ODP collection entity. Each manifestation is a member through the object property `collectionentity:isMemberOf` with the class `triple:Cluster`.

The ingestion and normalisation pipeline not only identifies duplicate documents but also performs numerous other data cleaning operations. Some metadata values, deemed unusable for search purposes within the GoTriple platform, are discarded. However, the discarding operation does not lead to their permanent removal but rather to their exclusion from the search systems. Therefore, Triple must model the possibility that certain entities may need to be marked as "discarded". Specifically, in this scenario, we consider the case of keywords. If a keyword is discarded, it is possible to specify an `xsd:boolean` through a data property `triple:isDiscarded`.

5.6 Scenario 6: Profile and User Account

This section discusses the representation of the Profile and User Account entities and the relationships that unite them.

5.6.1 Motivating Scenario

From each research artefact retrieved from external providers, the people or organisations present in the metadata are extracted. Each author is automatically saved as a unique entity within the platform. An illustrative case could be as follows: three research articles, written by the same author, are retrieved. Each author is saved in an Elasticsearch index as a unique entity, with its own ID, even though it is the same author. The system subsequently attempts to recognize the three authors as a single entity, corresponding to the identity of a single existing researcher. Therefore, a possible modelling approach must consider the following levels:

- *Profile*: Each person or organisation extracted from a research artefact retrieved from providers. This means there can be multiple author profiles corresponding to a single person.
- *Person or Organization*: The unique entity corresponding to an existing identity.

Moreover, Each profile corresponds to a name. The name, like keywords, is an entity that could be discarded.

Finally, the GoTriple platform also envisions that each person can be connected to a user account. Therefore, the modelling must also consider this entity, directly connected to a person identified from the research artefacts retrieved from data providers.

5.6.2 Competency Question

- Does the person who wrote the article have an associated account?
- How many profiles associated with this account have been retrieved?
- What articles were written by the person associated with this account?

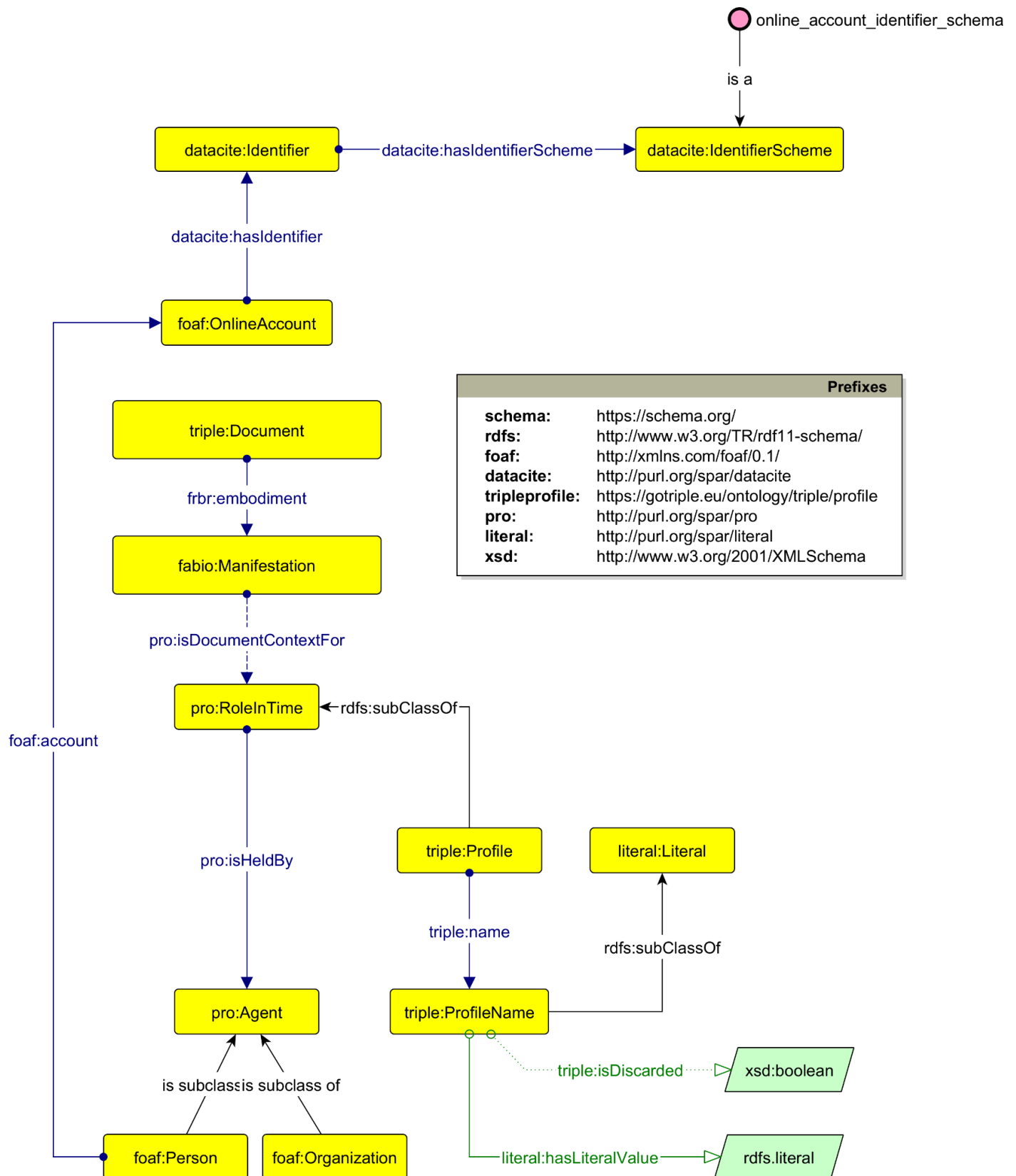


Figure 9: iteration 6.

5.6.3 Description

The materials for this iteration can be found in the directory located at <https://github.com/AlessandroBertozzi/TRIPLE-ontology/tree/main/development/06>. The entities depicted in the Graffoo image (Figure 9) represent the fundamental entities and relations of this iteration.

In this iteration, the modelling focuses on profiles, another central entity in GoTriple. Detailed decisions made during the modelling process are extensively covered in the previous chapter (see 4.3). The profile is represented by the class `triple:Profile`, a subclass of `pro:RoleInTime`. The `triple:Profile` class connects to the `triple:ProfileName` class, which extends the `literal:Literal` class. This class can specify an `rdfs:Literal` through the data property `literal:hasLiteralValue`. In addition, the name, as with keywords, will have a data property `triple:isDiscarded`.

Each `foaf:Person` can be associated with an account. To represent the account entity, we reuse the FOAF pattern `foaf:OnlineAccount`, connected to a `foaf:Person` via the object property `foaf:account`. Each `foaf:OnlineAccount` can have an associated identifier. As in previous iterations, we reused the pattern provided by the DataCite ontology. Therefore, each `OnlineAccount` will have an associated identifier linked to a `datacite:IdentifierSchema` named `triple:online_account_identifier_schema`.

5.7 Scenario 7: Project

In this iteration, the last fundamental entity of Triple is presented: projects. This iteration makes it possible to correctly represent the projects with all related metadata.

5.7.1 Motivating Scenario

In addition to documents, GoTriple facilitates the integration of projects within the Social Sciences and Humanities (SSH) domain from external sources. Although integration is limited to a select number of external data sources, each project is accompanied by the following metadata:

- *Identification*: Analogous to documents, it is feasible to associate one or more identifiers with a project.
- *Temporal Placement*: Metadata are available to denote the project's duration or period of execution.
- *Roles*: The roles involved in the organisation and funding of the project are specified.
- *Subject*: Similar to documents, a project can be linked to one or more disciplinary categories or to one or more keywords.
- *Name, Description*: Basic textual metadata are provided to offer human-readable information about the project.
- *Grant*: Each project can have multiple organisations or people funding the project.

5.7.2 Competency Question

- What organisations was the project funded by?
- What people and with what role are involved in the project?
- When did the project start?
- When is the project scheduled to end?
- What discipline is the project devoted to?

5.7.3 Description

The materials for this iteration can be found in the directory located at <https://github.com/AlessandroBertozzi/TRIPLE-ontology/tree/main/development/07>. The entities depicted in the Graffoo image (Figure 10) represent the fundamental entities and relations of this iteration.

The final iteration focuses on the project entity. The ontological representation of a project is similar to that of `triple:Document` (see sections 4.3 and 5.1). The main differences pertain to temporal placement, the definition of new roles (`pro:Role`), and the ability to specify the funding sources of the project.

The project is represented using the FOAF class `foaf:Project`. Temporal placement requires specifying a range, including a start date and an end date. To represent this date range, the TVC (Time Validity Concept) pattern is reused.

Three new `pro:Role` roles have been introduced: `triple:coordinating_entity`, `triple:sponsor`, and `triple:funder`.

Finally, for the funds allocated to the project, the Schema.org pattern `schema:Grant` is reused. The `foaf:Project` is connected to `schema:Grant` via the object property `schema:funding`.

Chapter 6: Discussion and conclusions

6.1 Discussion and Results

The Triple ontology was developed to meet various needs:

1. *Formalization of the Data Model with Semantic Standards*: Triple adopts a data model that conforms to international semantic standards, such as RDF (Resource Description Framework) and OWL (Web Ontology Language).
2. *Interoperability with External Sources*: Adopting an ontology that complies with semantic standards facilitates integration and interoperability with other external data sources. This is crucial for enabling the exchange of information between different systems and improving the quality and completeness of aggregated data.
3. *Reusability of Resources*: The Triple ontology is designed for the reusability of resources. The entities and properties defined in the ontology can be used in various contexts and applications, reducing the need to reinvent data structures for new projects and promoting consistency across different implementations.
4. *Documentation of Data Model Decisions*: Every decision made during the data model design is documented and justified within the ontology. This level of transparency is essential for ensuring that the choices are understood and shared among various stakeholders and for facilitating future modifications and improvements.
5. *Flexibility and Extensibility*: The structure of the Triple ontology is designed to be flexible and extensible. This allows it to be easily adapted to new knowledge domains or changes in project requirements without compromising the consistency of existing data.
6. *Facilitation of Collaboration and Knowledge Reuse*: A well-defined ontology promotes collaboration between different teams and organisations, facilitating the reuse and sharing of knowledge and contributing to the construction of a richer and more interconnected data ecosystem.

In this section, we will discuss whether these objectives have been achieved. Specifically, we will assess whether the Triple ontology aligns with the data models proposed by the aggregators mentioned in Chapter 2. These data aggregators made different choices in data modelling, leading

to divergences in the final results. To analyse these divergences, we propose some comparison tables between the data aggregators.

The first table (Table 13) evaluates the following points:

- *Available Documentation*: Investigates whether documentation or publications supporting the modelling decisions are available through the data aggregators' platforms.
- *Sharing of Controlled Vocabularies*: Assesses whether the aggregators provide controlled vocabularies for the values of certain fields. Sharing such vocabularies is crucial for studying and navigating the research materials made available through the data aggregators.
- *Available Data Model*: Checks whether the data model can be downloaded in any format.
- *Linking to External Resources*: In the context of Linked Data, investigates whether the data aggregator connects the collected resources to external resources (e.g., Wikidata) during the data enrichment phase.
- *Resources Described with URIs*: Examines whether the resources provided by the data aggregators are described with URIs.
- *Exporting Resources in a Semantic Web Standard*: Looks into whether the aggregator makes data available in standards aligned with those of the semantic web.

The following symbols will be used in the table:

Symbol	Meaning
✓	The element indicated in the column exists and is correctly documented.
O	The specified element is not handled by the considered data aggregator. Therefore, it cannot be returned.
-	The indicated element should exist, but it is not made available for consultation or reuse.
?	The indicated element is referenced by the data aggregator's documentation, but it is not properly made available or indicated.

Table 12: legend of table 13.

The second table (Table 14) addresses the extent to which the data models have reused existing standards outside their own data model definitions. To evaluate the degree of reusability, we use five levels:

- *None*: All entities and relationships are defined internally within the Data Model.
- *Weak*: Almost all entities and relationships are defined internally within the Data Model.
- *Good*: Some entities and relationships are defined internally within the Data Model.
- *Strong*: Almost all entities and relationships are defined externally to the Data Model.
- *Total*: All entities and relationships are defined externally to the Data Model.

Data Aggregator	Documentation is available	Vocabularies are available	Data Model is available	Resources URI	Linked to external resources	Using Semantic Web Standard
Europeana	✓	✓	✓	✓	✓	✓
OpenCitations	✓	O	✓	✓	O	✓
OpenAlex	✓	✓	✓	✓	✓	-
ro-Crate	✓	✓	✓	✓	✓	✓
ISIDORE	?	✓	?	✓	✓	?
OpenAIRE	✓	-	?	✓	✓	-
Triple	✓	✓	✓	✓	✓	✓

Table 13: Data Aggregator comparison: good practice coverage.

Data Aggregator	Reused Entities and Properties
Europeana	Weak
OpenCitations	Total
OpenAlex	None
ro-Crate	Strong
ISIDORE	Weak
OpenAIRE	None
Triple	Strong

Table 14: Data Aggregator comparison: entities and properties reused.

From the tables, it emerges that OpenCitations and RO-Crate are among the best examples of documented data models aligned with semantic web standards. In particular:

- *OpenCitations* reutilizes a significant portion of the SPAR Ontologies.
- *RO-Crate* largely reuses Schema.org for most of its entities and relationships, adding only a few relations not available in Schema.org.
- *Europeana* redefines the majority of its entities and relationships in an internal data model but reutilizes several widespread standards, such as RDF, for a small part of its entities and relationships.
- *ISIDORE* does not detail its data model extensively, except for the types of documents it makes available, which are mapped to an internal ontology and connected to terms in other vocabularies.
- *OpenAlex* defines its own internal mapping without reusing existing external entities but provides extensive and well-documented descriptions of its data model.

Most aggregators redefine their own data model, with limited reuse of existing data models. Except for OpenCitations and RO-Crate, the other aggregators prefer to define their own data modelling systems. The reasons are twofold. Internally, it is easier to manage data within their systems by redefining a standard that fits their specific needs. Externally, no sufficiently generic and comprehensive standards exist to cover the exhaustive data collected by these aggregators.

In Triple, we aimed to achieve results similar to those of OpenCitations and RO-Crate. GoTriple

was developed within a broader context, namely the TRIPLE project. The objective was to create an infrastructure for academic communication in the Social Sciences and Humanities (SSH). Therefore, a key goal in developing the Triple ontology was not only to adopt standards shared by scientific communities but also to make the decisions and actions taken documentable and shareable. Aligning with these two aggregators allowed Triple to meet the defined project objectives. Here, we briefly revisit the objectives mentioned at the beginning of this section in light of the results obtained during the modelling phase:

- *Describe the Data Model*: Triple comprehensively describes the GoTriple data model, adhering to the requirements defined with domain experts during the development phase (see Chapter 5). Additionally, it documents and makes the modelling decisions available through the publication of this document and the ontologies in HTML format. This increases the potential for reuse and sharing of the material described in this thesis.
- *URI for Resources*: Resources mapped by the Triple ontology are described using URIs.
- *Export in JSON-LD*: The platform allows resources to be exported in JSON-LD format.
- *Controlled Vocabularies*: Fields not available as free text but describable through a list of controlled values are made available as controlled vocabularies. To model them, a widely used standard like SKOS was reused. Furthermore, whenever possible, SKOS properties were reused to link Triple vocabularies to external vocabularies.

6.2 Limitations

Two significant limitations can be identified. The first is the lack of a more thorough reflection on the state of the art of other available data aggregators prior to modelling. The ontology definition, as described in Chapter 2, starts from the existence of an initial version of the data model. Therefore, the choices made were restricted to those already defined in that initial version. Connected to this first limitation is a second one: the ontology redefines its own internal system of data organisation. Although this approach aligns with the other data aggregators analysed throughout this paper (see Chapter 2.2 and Chapter 6.1), a limitation of reusability in different scenarios remains.

6.3 Further Developments

The work done for extending and refining the Triple ontology is continuing in ATRIUM (<https://atrium-research.eu/>), a four years EU funded research project.

The overarching objective of ATRIUM is to empower Arts and Humanities scholars in their use of digital methods by facilitating access to a wide range of reusable workflows and interoperable, composable services offered by leading research infrastructures in the Arts and Humanities domain. This includes improving the overall metadata quality of existing catalogues and repositories, enhancing multilingual support, and creating and disseminating workflows — potentially complex and non-linear sequences of steps — to describe how to perform specific tasks.

Specifically, ATRIUM focuses on aligning various ontologies within the Digital Humanities and Social Sciences domains, as addressed in tasks T3.2 and T3.1. This alignment seeks to improve metadata quality, a critical prerequisite for findability, by harmonising metadata curation and enrichment workflows between individual catalogues and their respective data sources. By strengthening the exchange between repository and catalogue providers, ATRIUM aims to optimise communication and align curation processes.

The focus of the GoTriple team in ATRIUM is, among others, to expand and integrate the Triple ontology to facilitate its interoperability with other widely used ontologies of the SSH and Cultural Heritage domain, including, but not limited to, AO-Cat (Felicetti et al., 2023), SSHOCro (Bekiari et al., 2022) and CIDOC CRM (<https://www.cidoc-crm.org/>). A preliminary experiment in this sense has been conducted by using the Mapping-Memory-Manager (3M) (<https://github.com/isl/Mapping-Memory-Manager>) tool using, for data integration and conversion, the X3ML language (Minadakis et al., 2015), a declarative, XML based language to describe schema mappings.

6.4 Conclusions

This thesis introduces Triple, an ontological model for representing research artefacts, projects, and researcher profiles, as part of the Triple project. Triple was designed based on a metadata analysis conducted on Deliverable D2.5 - Report on Data Enrichment, which contained an initial proposal of the ontology. The ontology was developed using SAMOD (Simplified Agile Methodology for Ontology Development), a data-driven, pattern-based methodology for ontology development. This approach allowed us to create a fully documented, extensible, and dynamic ontological model that accurately represents the material in question without excessive conceptual clutter, while also considering potential expansions to cover more information related to research in the social sciences and humanities.

The methodology proved efficient and effective for the task. The metadata analysis provided a comprehensive overview of the material and offered the researcher a solid foundation upon which the ontological model was built using SAMOD. The domain analysed is very broad, and the work

can be considered exhaustive for the requirements defined in collaboration with the domain expert within the Triple project.

ATRIUM aims to increase the level of interoperability with other data aggregators active in the SSH domain, starting with platforms like GoTriple. Therefore, it positions itself as an evolution of the project developed in this thesis.

References

Bibliography

Albertoni, Riccardo, David Browning, Simon J. D. Cox, Alejandra Gonzalez Beltran, Andrea Perego, and Peter Winstanley. "Data Catalog Vocabulary (DCAT) - Version 3." W3C, January 18, 2024. <https://www.w3.org/TR/2024/CR-vocab-dcat-3-20240118/>.

Aljalbout, Sahar, and Gilles Falquet. "A Semantic Model for Historical Manuscripts." Last modified 2018. <https://arxiv.org/abs/1802.00295>.

Aminu, Enesi, Ishaq Oyefolahan, Muhammad Abdullahi, and Muhammadu Salaudeen. "A Review on Ontology Development Methodologies for Developing Ontological Knowledge Representation Systems for Various Domains." *International Journal of Information Engineering and Electronic Business* 12 (2020): 28-39. <https://doi.org/10.5815/ijieeb.2020.02.05>.

Antezana, Erick, Martin Kuiper, and Vladimir Mironov. 2009. "Biological Knowledge Management: The Emerging Role of the Semantic Web Technologies." *Briefings in Bioinformatics* 10, no. 4: 392–407. <https://doi.org/10.1093/bib/bbp024>.

Arasteh-Roodsary, Sona Lisa, Emilie Blotiere, and Drahomira Cupar. "GoTriple, a European multicultural and multilingual gateway for the SSH research." *Seminar Arhivi, Knjižnice, Muzeji. Mogućnosti suradnje u okruženju globalne informacijske infrastrukture*, 2022.

Barker, Phil, and Lorna M. Campbell. "What is schema. org." *LRMI*. April 21, 2014. Accessed April 21, 2015.

Bekiari, Chrysoula, Athina Kritsotaki, Eleni Tsouloucha, and Maria Theodoridou. "D4.20 SSHOCro (Final Version)." April 2022. <https://zenodo.org/records/6771757>.

Brase, Jan. "DataCite-A Global Registration Agency for Research Data." In *2009 Fourth International Conference on Cooperation and Promotion of Information Resources in Science and Technology*. IEEE, 2009.

Breit, Anna, et al. "Combining Machine Learning and Semantic Web: A Systematic Mapping Study." *ACM Computing Surveys* 55, no. 14s (2023): 1–41.

Brickley, Dan, and Libby Miller. "FOAF Vocabulary Specification 0.91." 2007.

Brinkley, Dan, and R. V. Guha. 2004. "RDF Vocabulary Description Language 1.0: RDF Schema." W3C Recommendation.

Butt, Bilal H., Muhammad Rafi, and Muhammad Sabih. "A Systematic Metadata Harvesting Workflow for Analysing Scientific Networks." *PeerJ Computer Science* 7 (March 2021): e421. <https://doi.org/10.7717/peerj-cs.421>.

Canali, Daniela. "Web semantico e ontologie." *Biblioteche oggi* 23, no. 5 (2005): 50-58.

DataCite Metadata Working Group. *DataCite Metadata Schema Documentation for the Publication and Citation of Research Data*. Version 3.1. DataCite e.V., 2014. <http://doi.org/10.5438/0010>.

Date, Christopher John. *SQL and Relational Theory: How to Write Accurate SQL Code*. 2nd ed. Sebastopol, CA: O'Reilly Media, Inc., 2012.

David, Romain, et al. "Umbrella Data Management Plans to Integrate FAIR Data: Lessons from the ISIDORE and BY-COVID Consortia for Pandemic Preparedness." *CODATA Data Science Journal* 22 (2023): 35.

David, Sophie, Jean-Luc Minel, and Stéphane Pouyllau. "Documenting Some Uses of the Isidore Platform." 2011.

Daquino, Marilena, Arcangelo Massari, Silvio Peroni, and David Shotton. "The OpenCitations Data Model." *figshare* (2023). <https://doi.org/10.6084/M9.FIGSHARE.3443876>.

De Paoli, Stefano, et al. "Measuring and Promoting the Success of an Open Science Discovery Platform through 'Compass Indicators': The GoTriple Case." *Publications* 10, no. 4 (2022): 49.

De Nicola, Antonio, and Michele Missikoff. 2016. "A Lightweight Methodology for Rapid Ontology Engineering." *Communications of the ACM* 59, no. 3: 79–86. <https://doi.org/10.1145/2818359>.

De Nicola, Antonio, Michele Missikoff, and Roberto Navigli. 2005. "A Proposal for a Unified Process for Ontology Building: UPON." In *Database and Expert Systems Applications*, edited by Kim Viborg Andersen, John Debenham, and Roland Wagner, 655–64. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/11546924_64.

De Santis, Luca. "FAIR as a Journey: Lessons Learned and Takeaways from Building the GoTriple Discovery Platform for SSH." (2024).

Di Donato, Francesca, et al. "The Discovery Platform GOTRIPLE: An EOSC Service for Social Sciences and Humanities Research." In *AIUCD 2021-DH per la società: e-guaglianza, partecipazione, diritti e valori nell'era digitale* (2021): 31.

Doan, AnHai, Alon Halevy, and Zachary Ives. *Principles of Data Integration*. Elsevier, 2012.

Dumouchel, Suzanne, et al. "Addendum: Dumouchel, S., et al. GOTRIPLE: A User-Centric Process to Develop a Discovery Platform. Information 2020, 11, 563." *Information* 12.4 (2021): 166.

Dumouchel, Suzanne, et al. "GOTRIPLE: a user-centric process to develop a discovery platform." *Information* 11.12 (2020): 563.

Eirinaki, Magdalini, et al. "Recommender Systems for Large-Scale Social Networks: A Review of Challenges and Solutions." *Future Generation Computer Systems* 78 (2018): 413-418.

Falco, Riccardo, Aldo Gangemi, Silvio Peroni, David Shotton, and Fabio Vitali. 2014. "Modelling OWL Ontologies with Graffoo." In *The Semantic Web: ESWC 2014 Satellite Events*, edited by Valentina Presutti, Eva Blomqvist, Raphael Troncy, Harald Sack, Ioannis Papadakis, and Anna Tordai, 320–25. Lecture Notes in Computer Science. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-11955-7_42.

Felicetti, Achille, Carlo Meghini, Julian Richards, and Maria Theodoridou. "The AO-Cat Ontology." *Zenodo* (2023). <https://doi.org/10.5281/zenodo.7818375>.

Flanders, Julia, and Fotis Jannidis. 2015. "Data Modeling." In *A New Companion to Digital Humanities*, 229–37. John Wiley & Sons, Ltd. <https://doi.org/10.1002/9781118680605.ch16>.

Frigg, Roman, and Stephan Hartmann. 2006. "Models in Science." In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta. <https://plato.stanford.edu/archives/spr2020/entries/models-science/>.

Gandon, Fabien. "A survey of the first 20 years of research on semantic Web and linked data." *Revue des Sciences et Technologies de l'Information-Série ISI: Ingénierie des Systèmes d'information* (2018).

Gangemi, Aldo, Silvio Peroni, and Fabio Vitali. 2010. "Literal Reification." In *Proceedings of WOP 2010*, 65-66.

Gangemi, Aldo, and Valentina Presutti. 2009. "Ontology Design Patterns." In *Handbook on Ontologies*, edited by Steffen Staab and Rudi Studer, 221–43. International Handbooks on Information Systems. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-92673-3_10.

- Girvan, Michelle, and Mark E. J. Newman. "Community Structure in Social and Biological Networks." *Proceedings of the National Academy of Sciences* 99, no. 12 (2002): 7821-7826.
- Gomadam, Karthik, Peter Z. Yeh, and Kunal Verma. "Data Enrichment Using Data Sources on the Web." In *2012 AAAI Spring Symposium Series*. 2012.
- Guha, Ramanathan V., Dan Brickley, and Steve Macbeth. "Schema. org: Evolution of Structured Data on the Web." *Communications of the ACM* 59, no. 2 (2016): 44-51.
- Gruber, Tom. "What is an Ontology." 1993, 1-11.
- Guarino, Nicola, Daniel Oberle, and Steffen Staab. 2009. "What Is an Ontology?" In *Handbook on Ontologies*, edited by Steffen Staab and Rudi Studer, 1–17. International Handbooks on Information Systems. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-92673-3_0.
- Hitzler, Pascal. "A Review of the Semantic Web Field." *Communications of the ACM* 64, no. 2 (2021): 76-83.
- Hitzler, Pascal, et al. "OWL 2 Web Ontology Language Primer." W3C Recommendation, 27(1): 123, 2009.
- Hlupić, T., and J. Puniš. "An Overview of Current Trends in Data Ingestion and Integration." In *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, 1265-1270. Opatija, Croatia, 2021. <https://doi.org/10.23919/MIPRO52101.2021.9597149>.
- Hogan, Aidan. "The Semantic Web: Two Decades On." *Semantic Web* 11, no. 1 (2020): 169–85.
- Hyvönen, Eero. "Digital humanities on the semantic web: Sampo model and portal series." *Semantic Web* 14.4 (2023): 729-744.
- Hyvönen, Eero. "Using the Semantic Web in digital humanities: Shift from data publishing to data-analysis and serendipitous knowledge discovery." *Semantic Web* 11.1 (2020): 187-193.
- Iliadis, Andrew, et al. "One schema to rule them all: How Schema. org models the world of search." *Journal of the Association for Information Science and Technology* (2023).open
- Kassel, Gilles. "A Formal Ontology of Artefacts." *Applied Ontology* 5, no. 3-4 (2010): 223-246.
- Manghi, Paolo, et al. "OpenAIRE Research Graph Dataset." Dataset, Zenodo (2022). <https://doi.org/10.5281/zenodo.3516917>.

Maignien, Yannick. "ISIDORE, de l'interconnexion de données à l'intégration de services." 2011. <sic_00593320v2>.

McDaniel, Melinda, and Veda C. Storey. "Evaluating Domain Ontologies: Clarification, Classification, and Challenges." *ACM Computing Surveys* 52, no. 4 (July 2020): Article 70, 44 pages. <https://doi.org/10.1145/3329124>.

Meroño-Peñuela, Albert. 2013. "Semantic Web for the Humanities." In *The Semantic Web: Semantics and Big Data*, edited by Philipp Cimiano, Oscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, 645–49. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-38288-8_44.

Massari, Andrea, Francesca Mariani, Ignazio Heibi, Silvio Peroni, and David Shotton. "OpenCitations Meta." *Quantitative Science Studies* 5, no. 1 (2024): 50-75. https://doi.org/10.1162/qss_a_00292.

Miles, Alistair, and Sean Bechhofer. "SKOS Simple Knowledge Organization System Reference." W3C Recommendation, 2009.

Minadakis, N., Marketakis, Y., Kondylakis, H., Flouris, G., Theodoridou, M., de Jong, G., & Doerr, M. "X3ML Framework: An Effective Suite for Supporting Data Mappings." In *EMF-CRM@TPDL*, September 2015, 1-12.

Pastor-Sánchez, Juan-Antonio, Francisco Javier Martínez Méndez, and José Vicente Rodríguez-Muñoz. "Advantages of Thesaurus Representation Using the Simple Knowledge Organization System (SKOS) Compared with Proposed Alternatives." *Information Research: An International Electronic Journal* 14, no. 4 (2009): n4.

Patel, Archana, and Sarika Jain. "Present and future of semantic web technologies: a research statement." *International Journal of Computers and Applications* 43.5 (2021): 413-422.

Patel-Schneider, Peter F. "Analyzing schema. org." *The Semantic Web–ISWC 2014: 13th International Semantic Web Conference*, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I 13. Springer International Publishing, 2014.

Penteado, Bruno Elias, José Carlos Maldonado, and Seiji Isotani. "Methodologies for publishing linked open government data on the web: a systematic mapping and a unified process model." *Semantic Web* 14.3 (2023): 585-610.

Peroni, Silvio. "A Simplified Agile Methodology for Ontology Development." In *OWLED*, 2016. https://doi.org/10.1007/978-3-319-54627-8_5.

Peroni, S. (2017). A Simplified Agile Methodology for Ontology Development. In Proceedings of the 13th OWL: Experiences and Directions Workshop and 5th OWL reasoner evaluation workshop (OWLED-ORE 2016). https://doi.org/10.1007/978-3-319-54627-8_5

Peroni, Silvio, and David Shotton. "OpenCitations, an Infrastructure Organization for Open Scholarship." *Quantitative Science Studies* 1, no. 1 (2020): 428-444. https://doi.org/10.1162/qss_a_00023.

Peroni, S., Shotton, D. (2018). The SPAR Ontologies. In Proceedings of the 17th International Semantic Web Conference (ISWC 2018): 119-136. DOI: https://doi.org/10.1007/978-3-030-00668-6_8

Peroni, Silvio, David Shotton, and Fabio Vitali. "Scholarly Publishing and Linked Data: Describing Roles, Statuses, Temporal and Contextual Extents." In *Proceedings of the 8th International Conference on Semantic Systems*, 9–16. I-SEMANTICS '12. Graz, Austria: Association for Computing Machinery, 2012. <https://doi.org/10.1145/2362499.2362502>.

Peroni, Silvio, David Shotton, and Fabio Vitali. 2012. "The Live OWL Documentation Environment: A Tool for the Automatic Generation of Ontology Documentation." In *Knowledge Engineering and Knowledge Management*, edited by Annette ten Teije, Johanna Völker, Siegfried Handschuh, Heiner Stuckenschmidt, Mathieu d'Acquin, Andriy Nikolov, Nathalie Aussenac-Gilles, and Nathalie Hernandez, 398–412. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-33876-2_35.

Philippe Bourdenet. "L'espace documentaire en restructuration: l'évolution des services des bibliothèques universitaires." Le serveur TEL (thèses-en-ligne), tel-00932683.

Pokorný, Jaroslav. "Conceptual and Database Modelling of Graph Databases." In *Proceedings of the 20th International Database Engineering & Applications Symposium*, 2016.

Pouyllau, Stéphane et al., "Bilan 2011 de la plateforme ISIDORE et perspectives 2012-2015", MoDyCo, Modèles, Dynamiques, Corpus - UMR 7114, 10670/1.bqexsj.

Pouyllau, Stéphane, Laurent CAPELLI, Jean-Luc MINEL, Mélanie BUNEL, Nicolas SAURET, Olivier BAUDE, Hélène JOUGUET, Pauline BUSONERA, and Adrien DESSEIGNE. 'ISIDORE a 10 ans', 25 October 2021. <https://zenodo.org/records/5699997>.

- Pouyllau, Stéphane. "ISIDORE : reprise des mises à jour ! Carnet de recherche d'Huma-Num." 2023.
- Pouyllau, S., L. Capelli, J.-L. Minel, M. Bunel, N. Sauret, O. Baude, H. Jouguet, P. Busonera, and A. Desseigne. "ISIDORE a 10 ans." *Zenodo*, 2021. <https://doi.org/10.5281/zenodo.5699997>.
- Priem, J., Piwowar, H., & Orr, R. (2022). *OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts*. ArXiv. <https://arxiv.org/abs/2205.01833>
- Rejeb, A., et al. "The Big Picture on Semantic Web and Interoperability. What We Know and What We Don't." 2021.
- Rejeb, Abderahman, et al. "Charting Past, Present, and Future Research in the Semantic Web and Interoperability." *Future Internet* 14, no. 6 (2022): 161.
- Ronallo, Jason. "HTML5 Microdata and Schema.org." *Code4Lib Journal* 16 (2012).
- Sattar, Abdul, et al. "Comparative Analysis of Methodologies for Domain Ontology Development: A Systematic Review." *International Journal of Advanced Computer Science and Applications* 11, no. 1 (2020): 1-10. <https://doi.org/10.14569/IJACSA.2020.0110101>.
- Sefton, Peter, et al. "RO-Crate 1.1." Researchobject.org community, April 26, 2023. <https://doi.org/10.5281/zenodo.7867028>. <https://w3id.org/ro/crate/1.1>.
- Shimizu, Cogan, Karl Hammar, and Pascal Hitzler. "Modular Ontology Modeling." *Semantic Web* 14, no. 3 (2023): 459-489.
- Shotton, David, and Silvio Peroni. "The DataCite Ontology." Version 1.2, September 15, 2022. <http://purl.org/spar/datacite>. Contributors: Amy J. Barton, Egbert Gramsbergen, Jan Ashton, and Marie-Christine Jacquemot. Imported Ontologies: <http://purl.org/spar/literal>. Distributed under a Creative Commons Attribution 4.0 International License.
- Silberschatz, Abraham, Henry F. Korth, and Shashank Sudarshan. *Database System Concepts*. 5th ed. New York: McGraw-Hill, 1997.
- Spinaci, Gianmarco, Giovanni Colavizza, and Silvio Peroni. "A Map of Digital Humanities Research across Bibliographic Data Sources." *Digital Scholarship in the Humanities* 37, no. 4 (December 2022): 1254–1268. <https://doi.org/10.1093/lhc/fqac016>.
- Tabassum, Shazia, et al. "Social Network Analysis: An Overview." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, no. 5 (2018): e1256.

Teorey, Toby J., Dongqing Yang, and James P. Fry. 1986. "A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model." *ACM Computing Surveys (CSUR)* 18, no. 2: 197–222. <https://doi.org/10.1145/7474.7475>.

Uschold, Mike, and Michael Gruninger. 1996. "Ontologies: Principles, Methods and Applications." *The Knowledge Engineering Review* 11, no. 2: 93–136. <https://doi.org/10.1017/S0269888900007797>.

Vera-Olivera, Harley, Guo Ruizhe, Maristela Holanda, Ruben Cruz Huacarpuma, Ana Paula Bernardi da Silva, and Ari Melo Mariano. "Data Modeling and NoSQL Databases - A Systematic Mapping Review." *ACM Computing Surveys* 54, no. 6, Article 116 (July 2021): 26 pages. <https://doi.org/10.1145/3457608>.

Weigand, Hans, Paul Johannesson, and Birger Andersson. "An Artifact Ontology for Design Science Research." *Data & Knowledge Engineering* 133 (2021): 101878.

Weigand, Hans, and Paul Johannesson. "How to Identify Your Design Science Research Artifact." In *2023 IEEE 25th Conference on Business Informatics (CBI)*. IEEE, 2023.

Wood, David, Marsha Zaidman, Luke Ruth, and Michael Hausenblas. *Linked Data*. Manning Publications Co., 2014.

Wu, Jia, Zhigang Chen, and Ming Zhao. "Community Recombination and Duplication Node Traverse Algorithm in Opportunistic Social Networks." *Peer-to-Peer Networking and Applications* 13 (2020): 940-947.

Yang, Wanting, et al. "Semantic Communications for Future Internet: Fundamentals, Applications, and Challenges." *IEEE Communications Surveys & Tutorials* 25, no. 1 (2022): 213–250.

Yu, Liyang, and Liyang Yu. "Schema.org and Semantic Markup." In *A Developer's Guide to the Semantic Web* (2014): 475–515.

Sitography

"Agent, n." *OED Online*. March 2024. Oxford University Press. https://www.oed.com/dictionary/agent_n1?tab=factsheet#8694696.

Berners-Lee, Tim. 2006. "Linked Data - Design Issues." Accessed June 10, 2024. <https://www.w3.org/DesignIssues/LinkedData.html>.

Berners-Lee, Tim. 2011. "Tim Berners-Lee on the Next Web." TED Video. April 10, 2011. https://web.archive.org/web/20110410204952/http://www.ted.com/talks/tim_berners_lee_on_the_next_web.html.

"Data Aggregation." Hevo Data. Accessed June 12, 2024. <https://hevodata.com/learn/data-aggregation/>.

"Data Aggregators." Factorial Digital. Accessed June 12, 2024. <https://factorialdigital.com/data-aggregators/>.

Europeana. n.d. "EDM Documentation." Accessed June 2, 2024. <https://pro.europeana.eu/page/edm-documentation>.

Europeana. 2016. EDM Definition v5.2.7. April 2016. Accessed May 5, 2024. https://pro.europeana.eu/files/Europeana_Professional/Share_your_data/Technical_requirements/EDM_Documentation/EDM_Definition_v5.2.7_042016.pdf.

GoTriple. "A Central Access Point for the Social Sciences and Humanities." *PUBMET*, 2022, 26-27. <https://doi.org/10.15291/pubmet.3927>.

Huma-Num. "ISIDORE Documentation." Accessed June 12, 2024. <https://documentation.huma-num.fr/en/isidore-en/>.

ISIDORE. "ISIDORE Vocabularies." Accessed June 12, 2024. <https://isidore.science/vocabularies>.

ISIDORE Vocabularies. 2024. "ISIDORE Vocabularies." Retrieved June 9, 2024. <https://isidore.science/vocabularies>.

ISIDORE Documentation. 2024. "ISIDORE Platform Documentation." Retrieved June 9, 2024. <https://documentation.huma-num.fr/isidore/>.

"Linked Data Platform." 2024. In Wikipedia. June 2, 2024. https://en.wikipedia.org/w/index.php?title=Linked_Data_Platform&oldid=1226948186.

OpenAlex. 2024. "OpenAlex Technical Documentation." May 9, 2024. <https://docs.openalex.org>.

"Schema.Org - Schema.Org." Accessed June 12, 2024. <https://schema.org/>.

