I. J. Education and Management Engineering, **2026, 4, 1-15**
Published Online on August 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijeme.2026.04.01

Ontological Cores of Documents: A Construction Methodology and Use Cases

Mohammed El Ouaazizi
Laboratory of Engineering Sciences (LSI), Polydisciplinary Faculty of Taza, USMBA University, Morocco
E-mail: mohammed.elouaazizi@usmba.ac.ma
ORCID iD: https://orcid.org/0009-0005-5013-2225

Received: April 14, 2026; Revised: May 18, 2026; Accepted: June 17, 2026; Published: August 8, 2026

Abstract: The preservation of rare documents in the form of image collections presents significant challenges regarding access to their documentary content. To enable this accessibility for software agents, this article proposes a formal representation of such documents through a semantic description layer. This layer includes a set of descriptive metadata attached to the document, alongside the minimal vocabulary required to formalize the explicit textual and visual knowledge of its documentary content. To achieve this, we present a construction methodology based on a Semantic Model of Document (SMD), where a document is treated as a core documentary resource composed of a set of information resources. The semantic description of these resources, aligned with RDF framework logic, produces an Ontological Core of Document (OCD) that formally describes the document's logical structure and captures its underlying semantics. Finally, we demonstrate the practical utility of these Ontological Cores through three distinct use cases—each targeting a specific dataset level (structural, administrative, and semantic)—showing how they allow software applications to move beyond simple collection searching toward intelligent, precise information extraction directly from the documentary content.

Index Terms: Digitized Document, Semantic Model, Structural Ontology, Core Ontology, Ontological Core, Knowledge Engineering, Digital Libraries, Semantic Web

1. Introduction

Digitization is a widely used practice for the digital preservation of rare documents, such as ancient manuscripts of cultural heritage. It produces collections of images that can be stored and described by bibliographic records following specific rules to facilitate their management and retrieval [1].

This article is not limited solely to the issue of cataloging or indexing these collections; its main motivation is to make their documentary content accessible and usable by software agents to promote their large-scale utilization [2]. To achieve this goal, we follow a semantic modeling approach for digitized documents based on the Resource Description Framework (RDF) to provide a formal representation of both the logical structure of digitized documents and the semantics of the information expressed in their content. This approach consists of defining a Semantic Model of Document (SMD) and its subsequent formalization using semantic web languages [3].

In the SMD model, a document is represented as a collection of images. Within these images, critical document fragments are identified; while the majority of these fragments express content in natural language, they collectively convey the explicit textual and visual knowledge embedded within the document. The digitized document itself is treated as a core documentary resource, uniformly identifiable by its unique Uniform Resource Identifier (URI). Concurrently, the relevant document fragments contained within it are modeled as distinct information resources, each uniformly identifiable relative to the base URI of the parent documentary resource.

All resources attached to a document are described using the RDF data model. Their structural properties are defined using a Structural Ontology of Document (SOD) that we designed and built specifically for the structural description of digitized document image collections. In addition, the administrative description of a digitized document is carried out using metadata from the Dublin Core (DC) vocabulary for cataloging purposes, while the semantic description of relevant document fragments is expressed using a Core Ontology of Document (COD), created by specializing an Upper-Level Ontology (ULO) with the necessary concepts and semantic relationships. All the descriptive data associated with the document, as well as its COD, constitute an Ontological Core of Document (OCD).

Unlike traditional document management systems that rely on standard keyword indexing [5] or low-level image descriptors [6], our approach addresses a critical qualitative and technical gap. Standard pipelines involving character

recognition (OCR) followed by Natural Language Processing (NLP) often struggle when dealing with cultural heritage manuscripts due to severe handwriting irregularities [9, 10]. By defining a specialized SOD ontology and a lightweight COD ontology, our methodology allows the generation of an OCD knowledge base. Acting as a comprehensive document-centric knowledge base (KB), each OCD encapsulates these ontological schemas alongside Dublin Core metadata, integrating their respective structural, administrative, and semantic datasets. This bridges the gap between flat metadata templates and heavyweight, non-scalable domain ontologies, establishing a novel benchmark for the semantic exploitation of digitized image collections.

The remainder of this article is organized as follows: Section 2 addresses the core problem statement and motivations guiding our decentralized semantic approach. Section 3 delineates the abstract Semantic Model of Document (SMD) framework along with its standardized architectural nomenclature. Section 4 outlines the underlying technical infrastructure and W3C web standards required for its machine-processable formalization. Section 5 details the formal knowledge engineering process of the Ontological Cores of Document (OCD), explicitly formalizing the interplay between the Structural Ontology (SOD) and the Core Ontology (COD). Section 6 introduces the operational algorithms and hybrid co-construction pipelines that govern graph population. Section 7 presents three distinct application use cases validating structural, administrative, and semantic datasets. Section 8 provides empirical validation metrics and performance benchmarks derived from our prototype deployment. Section 9 discusses current limitations and outlines future work before Section 10 concludes the paper.

2. Problematic and Motivations

The contemporary Web remains predominantly a collection of documents whose content was originally designed to be consumed exclusively by humans. Consequently, the role of machines is limited to rendering this content to users and facilitating basic navigation between documents via traditional hyperlinks [4].

Within this architectural paradigm, search engines serve as the primary tools for information retrieval. These systems index documents based on the statistical weight of significant words appearing in their textual content. Accordingly, responses to keyword-based user queries are evaluated solely through the prism of lexical similarity between the query terms and the pre-computed document indexes [5].

Documents stored in image format introduce a significantly higher level of complexity regarding content accessibility. Unlike text, images do not inherently rely on a functional, structurally manipulable language system. To circumvent this, search engines typically index image-based documents using low-level visual descriptors such as color histograms, textures, and geometric shapes. Image retrieval is therefore restricted to visual similarity matches between a query image and indexed repository images [6].

Document retrieval governed by such keyword or visual similarity matching frequently results in long lists of hyperlinks. Users are then forced to manually browse through numerous files to evaluate their actual relevance. To substantially enhance retrieval performance, document content must be indexed at a semantic level, moving beyond surface-level keywords or low-level physical attributes [7].

While the semantic indexing of textual content generally leverages Natural Language Processing (NLP) tools for automated comprehension, the limitations and computational costs of full-text NLP have led to more pragmatic alternatives. Modern approaches focus on extracting isolated, high-value anchors of meaning, such as named entities [8].

However, in our specific context, historical documents are digitized as image collections containing complex, interleaved layouts: textual regions combined with non-textual representations like illustrations and graphics. Extracting raw text from these historical manuscript images using standard optical character recognition (OCR) or handwriting text recognition (HTR) faces severe bottlenecks due to the inherent irregularity of ancient handwriting [9]. Consequently, cascading full-text extraction and subsequent semantic processing constitute a highly complex, error-prone, and imprecise pipeline for operational environments [10]. Recent advances in deep learning for Handwritten Text Recognition (HTR) have improved raw text extraction, yet bridging the gap between irregular historical layouts and formal knowledge graphs remains an open challenge requiring tightly coupled spatial-semantic models [25].

To resolve these limitations, we shift the focus away from traditional full-text indexing. Our objective is to deliver a rich, formal, and highly expressive semantic description of digitized manuscripts while strictly preserving their original image-based nature. Our approach overcomes the barrier of content accessibility by introducing a semantic modeling framework centered around specialized, localized Ontological Cores generated individually per document. By explicitly separating concerns into an integrated, modular architecture, these lightweight cores remain fully computable, accessible, and reusable by semantic web software agents to execute diverse advanced application tasks. This aligns with contemporary shifts in cultural heritage digitization, which increasingly favor modular, decentralized ontology networks and localized knowledge graphs over monolithic, single-schema repositories to reduce semantic friction and computational reasoning overhead [24, 26, 27].

To clearly delineate the scientific and technical contributions of our framework against the challenges identified in existing literature, Table 1 provides a structural comparison based on key operational and architectural features.

Table 1. Comparative analysis of the proposed framework against existing document modeling paradigms.

Feature / Criteria	Monolithic Domain Ontologies	TEI / XML-Based Approaches	Standards-Based Aggregators (e.g., Europeana / EDM)	Our Proposed Framework (SMD / OCD)
Architecture Scale	Global / Monolithic	Document / File level	Repository / Monolithic	Individualized per Document (Modular OCD)
Granularity Level	Global Document	Textual Line / Element	Record / Item Level	Document fragment Level
Knowledge Evolution	Static (Pre-defined schema)	Rigid (Fixed Schema/DTD)	Static (Fixed Properties)	Dynamic (Progressive Co-construction)
Reasoning & Graph Mass	High (Heavy global axioms)	None (Syntax only)	Medium (Flat Taxonomies)	Very Low (Lightweight localized COD)
Spatial-Semantic Link	Absent	Implicit (Text flow)	Weak	Explicit (Interleaved via Graph)

3. Semantic Model of Document

The digital preservation of cultural heritage manuscripts in image format ensures long-term conservation but introduces two major barriers to automatic semantic exploitation by software agents:

- **The Semantic Gap:** Document images are natively processed via low-level visual attributes (color, texture, shape), which lack any direct connection to the underlying meaning of the contained information [11].
- **Content Informality**: The encyclopedic and unstructured nature of historical documentary content—reflecting the author's unique discourse style—creates high complexity for automated knowledge extraction [12].

To bridge this gap, we propose the Semantic Model of Document (SMD). Built upon the RDF data model, the SMD formalizes both the physical/logical structure of digitized collections and the explicit knowledge embedded within their content [13]. RDF's flexible triple structure (subject, predicate, object) enables the incremental assembly of simple statements into rich, interconnected graph descriptions.

Core to the SMD is a three-tiered conceptual hierarchy that maps the document from its physical digital footprint to its semantic components:

- **Document**: The root entity representing the digitized work as a whole, uniformly identified by a unique base URI.
- **Image**: The operational layout unit, where each image corresponds to a single digitized page of the parent document.
- **Fragment**: The semantic unit, identifying specific regions of interest within a page (e.g., text blocks, tables, illustrations) that encapsulate the explicit textual and visual knowledge of the discourse domain.

These core concepts (Document, Image, Fragment) and their real-world instances are interconnected via structural, administrative, and semantic attributes, fully formalized in compliance with RDF logic (Fig. 1).

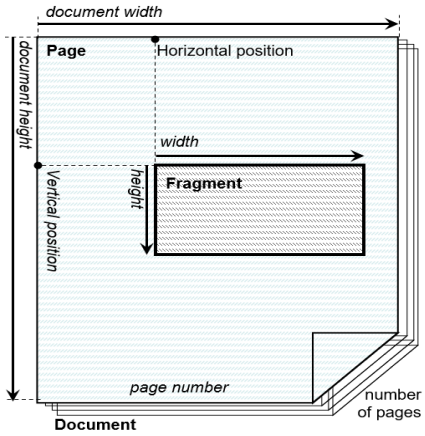

Fig. 1. Logical structure of a digitized document

Given the multi-layered architecture driven by this model, and to eliminate any conceptual ambiguity or overlapping terminology between document engineering and semantic web frameworks, Table 2 provides the standardized nomenclature used throughout the remainder of this manuscript.

Table 2. Standardized architectural nomenclature and glossary

Standardized Term	Acronym	Exact Scope & Architectural Role	Prohibited/Mixed Terms (To Avoid)
Ontological Core of Document	OCD	The global, individualized multi-layered graph instance generated for each single document (comprising the DC, SOD, and COD layers).	Core Ontological, Ontological Core
Core Ontology of Document	COD	The specific semantic layer within the OCD that contains domain-specific knowledge and extracted concepts of meaning.	Ontology Core, Core Ontology
Structural Ontology of Document	SOD	The specific structural layer within the OCD that maps physical coordinates (x, y, width, height) of document zones.	Spatial Ontology, Spatial Core
Semantic Model of Document	SMD	The overarching conceptual framework and abstract pipeline proposed in this paper to govern OCD generation.	Document Semantic Model
Document fragment	FRG	A localized, physical region of interest isolated on a manuscript page carrying a specific administrative, structural, or semantic unit.	Fragment, Piece, Unit

4. SMD Formalization and Infrastructure

To ensure that the SMD model is machine-processable and interoperable, its conceptual entities are formalized using W3C Semantic Web standards. Rather than employing natural language descriptions, all domain knowledge is structured as unambiguous, directed semantic graphs compiled via the RDF framework.

4.1. Resource Identification (URIs)

Within the SMD framework, every physical and logical entity—specifically the digitized document, digitized page, and document fragment—is treated as an independent Semantic Web resource. To ensure global uniqueness, persistent web resolution, and proper network addressability, each resource is assigned a strict hierarchical HTTP-URI pattern as detailed in Table 3.

Table 3. Hierarchical HTTP-URI patterns for SMD resources

Resource	Formal HTTP-URI Pattern
digitized document	**http://host:port/domain/docId**
digitized page	**http://host:port/domain/docId/imgId**
document fragment	**http://host:port/domain/docId/frgId**

Any secondary or auxiliary metadata generated during the document lifecycle is systematically appended using relative sub-URIs branched directly from the document's root identifier (base URI).

4.2. Schema and Axiomatization

The modeling approach transitions from a flat asset registry to a computable knowledge network by leveraging RDF-Schema (RDFS) and Web Ontology Language (OWL) primitives. While RDFS provides the baseline class taxonomy and basic properties constraints (defining rdfs:domain and rdfs:range), OWL-DL is enforced to inject advanced axiomatization. This ontological layer guarantees strict cardinality constraints, property characteristics, and formal logic boundaries required to safely specialize the ULO ontology into our custom domain schemas without schema drift.

4.3. Serialization, Storage, and Querying

The underlying graph structure forms a labeled directed multigraph serialized in standard RDF/XML syntax, ensuring native compatibility with XML manipulation parsers. Long-term data persistence and transactional updates are handled via a dedicated Triplestore architecture. Data accessibility and machine-to-machine application integration are decoupled through a standard SPARQL endpoint. This querying interface enables software agents to evaluate the graph, perform pattern-matching graph traversals, and dynamically retrieve explicit or inferred semantic triples.

5. Engineering of the *Ontological Cores of Document*

The formalization of the SMD model produces an Ontological Core of Document (OCD). Characterized as a self-contained, document-centric Knowledge Base (KB), each OCD encapsulates both the terminological schemas (TBox) and their corresponding empirical assertions (ABox) (Fig. 2). Rather than being a static template, the OCD is engineered through a continuous, progressive enrichment process that synthesizes structural, administrative, and domain-specific semantic layers. Mechanically, the architectural breakdown of an OCD is formalized as follows:

$$OCD=\{\ TBox_{Schema}\ \}\cup\{\ ABox_{Datasets}\ \}$$

The Ontological Schema Layer (TBox): *provides the foundational vocabulary* and formal constraints through three unified components:

- **Dublin Core (**DC**) Vocabulary**: Standard properties for global resource identification and cataloging.
- **Structural Ontology of Document (**SOD**)**: *Our custom ontology* derived directly from the SMD model to formalize document layout.
- **Core Ontology of Document (**COD**):** A lightweight schema generated from scratch and strictly specialized from a ULO.

***The Knowledge Population Layer* (*ABox*)**: A dynamic dataset progressively enriched during the document analysis lifecycle:

- **Administrative Dataset**: Metadata populated via DC for bibliographic cataloging.
- **Structural Dataset**: Formed through the automatic instantiation of the SOD schema, capturing pages, dimensions, and layout coordinates.
- **Semantic Dataset**: Generated by analyzing document fragments, translating informal visual/textual content into explicit domain instances, while simultaneously feeding missing conceptual classes and relationships back into the COD.

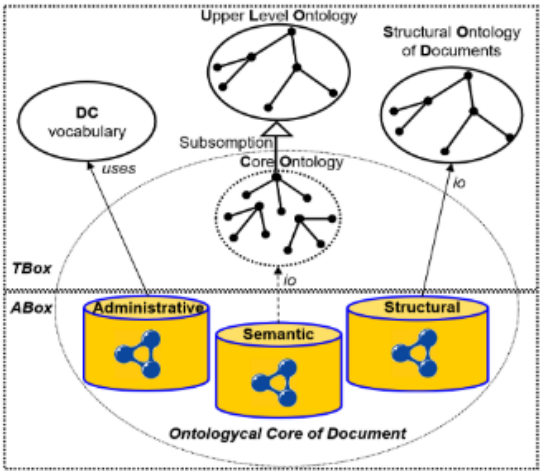

Fig. 2. Ontological Core of Document (OCD)

5.1. Administrative Description of the Container

Administrative metadata uses the standard HTTP-URI pattern (http://host:port/docId) to ensure web scalability. The baseline description leverages the 15 core elements of the Dublin Core (DC) vocabulary [15], ensuring interoperability across web catalogs without replacing deep archival formats (e.g., MARC, EAD) [16]. Extended descriptive properties utilize the DC Terms and Creative Commons (CC) namespaces to handle fine-grained licensing and metadata enrichment.

5.2. Structural Description

The SMD model defines Document, Image, and Fragment as essential concepts corresponding to all types of real objects that we can identify in a digitized document in form of a collection of images. These objects are defined as follows:

- **Digitized document** is a documentary resource of the Document type. It is a collection of images that is created by scanning the pages of an original document. It is characterized by its format, which corresponds to the size of any image in the collection, and by the number of images that make this collection.
- **Digitized page** is a type of image that is part of a collection of images and is considered a particular document. It is characterized by a page number that indicates its order in the collection of images.
- **Document fragment** is an information resource of a Fragment type. It is obtained by extracting a part of an image from a collection. It is characterized by its size and its relative position in relation to the image from which it was extracted. We can distinguish several types of document fragments based on how they present the information (TextArea, Table,…).

We have organized all concepts into a taxonomy of classes, characterized by the data properties and linked by the semantic relationships as explained in the previous section, and built a Structural Ontology of Document (SOD) “Fig. 3,”. This ontology provides the necessary vocabulary to express all the structural characteristics of resources attached to the modelled document as well as all the existing spatial relations between these resources [14].

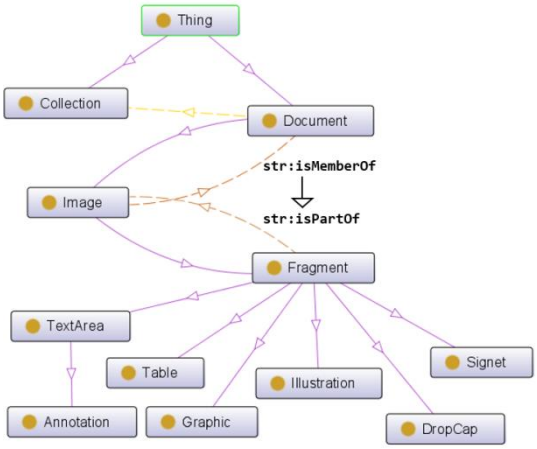

Fig. 3. Structural Ontology of Document (prefix str:<http://host:port/voc/str.owl>)

Any resource identified at the document level is necessarily an instance of one of the classes of the SOD ontology. It is characterized by properties Table 4, called structural attributes, and semantic relationships that define it within the logical structure of the document.

Table 4. Structural concepts and attributes

Concept	Object	Resource Type	Definition	Structural Attributes
Document	digitized document	Documentary Resource	A collection of digitized images representing the original work.	width, height, numberOfPages.
Image	digitized page	Operational Layout	A single digitized page within the collection order.	pageNumber
Fragment	document fragment	Information Resource	An extracted sub-region of an image (e.g., TextArea, Table).	width, height, horizontalPosition, verticalPosition.

To handle complex interleaved layouts (e.g., marginal glosses), the SOD ontology integrates two topological properties: str:isContiguousTo and str:isOverlappedBy. These specific predicates map spatial-semantic intersections directly within the individual graph, allowing localized reasoning algorithms to preserve the manuscript's original context.

The structural description of all the resources attached to a modelled document uses the SOD ontology as a formal model for the automatic production of all the necessary structural data for the extraction of representations of these resources in the form of image fragments.

5.3. Semantic Description of Content

A modeled document comprises a set of particularly important document fragments that express, informally and in varying degrees of detail, the central concepts and objects of its content. Identifying these fragments is a delicate operation that requires a thorough understanding of the main ideas underlying the document's cognitive foundation.

The semantic richness and encyclopedic nature of written cultural heritage, particularly ancient manuscripts, necessitates the use of a multitude of formal vocabularies and ontologies for their semantic description. This is why our approach proposes to associate with each modeled document a Core Ontology of Document (COD) which provides the minimal and strictly necessary vocabulary for its semantic description.

The COD is built from scratch, then enriched during the OCD construction process using an evolutionary prototyping strategy with a middle-out approach to build stable taxonomies [17]:

- **Taxonomy Evolution**: Concepts and properties discovered within fragments are progressively appended to the ontology schema.
- **Strict Specialization**: To enforce semantic consistency and prevent structural drifting, all custom concepts and properties are created as sub-classes (rdfs:subClassOf) or sub-properties (rdfs:subPropertyOf) of a foundational Upper-Level Ontology (ULO) [18].
- **Property Signatures**: Relationships are strictly constrained by defining their precise semantic scope through Domain (rdfs:domain) and Range (rdfs:range) assertions.

5.4. Proposed Construction Methodology

Based on the administrative, structural, and semantic layers detailed in the previous subsections, this section outlines our overall construction methodology. Building the Ontological Core of Document (OCD) is an integrated, multi-stage engineering process where the COD ontology and the three descriptive datasets (ABoxes) evolve together [19]:

- **In the first stage**, the document's basic administrative metadata (e.g., title, author(s)) and its high-level structural data (e.g., total number of pages, layout dimensions) are automatically instantiated (using the DC vocabulary and the **SOD** ontology) to form the document container.
- **The second stage**, consists of an iterative process of semantic analysis of the document fragments.

Formally, each identified document fragment is modeled as an information resource uniformly identified by its relative URI<http://host:port/docId/frgId>. Its structural properties (width, height, and relative spatial coordinates of the parent image) are automatically added to the structural dataset, while its semantic analysis by the human expert triggers a dual intertwined enrichment pipeline [20]: the conceptual constraints of the COD ontology are extended precisely when a new concept or relationships is discovered, while the semantic datasets are enriched precisely when a new instance is discovered. Guided by the structural options detailed in Table 5, this joint construction follows two synchronized operational workflows:

Field A: Schema Enrichment Layer (TBox / COD Evolution)

When a fragment introduces new domain knowledge, it dictates the structural evolution of the Core Ontology of Document (COD) through three strict taxonomic rules:

- **Concept Creation (2)**: If a fragment introduces a domain concept absent from the COD, the concept is immediately created by specializing (rdfs:subClassOf) either an existing COD class or a foundational class from the ULO [18].
- **Property and relationship discovery (3):** Analyzing a fragment's content may reveal novel semantic relationships connecting domain concepts. These object or data properties are appended to the COD ontology, either directly or by specializing (rdfs:subPropertyOf) an existing COD relationship or a generic ULO relationship.
- **Signature Constraint Enforcement (2, 3)**: To preserve ontological consistency, every newly discovered relation must have a strictly defined signature. If a concept required to declare the domain (rdfs:domain) or co-domain (rdfs:range) of a relationship does not exist yet in the COD, it must be programmatically created (2) before the relation itself can be safely appended (3).

Field B: Knowledge Population Layer (ABox / Semantic Dataset Enrichment)

Concurrently, operational fragments are leveraged to populate the OCD knowledge base through two instantiation rules and one annotation rule:

- **Instance Instantiation and Valuation (4)**: If a fragment captures a concrete real-world entity, the corresponding resource is asserted as an instance (rdf:type) of its designated COD concept class. If the designated concept class does not yet exist, it is generated on the fly via the TBox workflow (2).
- **Semantic Linkage Populating (3, 4)**: The inner content of an instance fragment is parsed to extract its specific characteristic attributes (Data Properties) and its semantic linkages with other resources (Object Properties). These are used to richly describe the target resource (4). If any supporting concept (2, 3) or related target resource is missing from the graph during this step, they are recursively generated first.
- **Conceptual Annotation (1)**: If a fragment explicitly represents or discusses a concept already formalized in the COD ontology, the concept class itself is bound directly to the fragment's URI via the rdfs:seeAlso property, anchoring informal visual text directly to its formal definition.

Table 5. Different possibilities for enriching the OCD

N°	RDF/XML Syntax Examples
(1)	<rdf:Description about="http://host:port/docId/ClsId"> <rdfs:seeAlso rdf:resource="http://host:port/docId/frgId"/> </rdf:Description>
(2)	<rdfs:Class rdf:about="http://host:port/docId/ClsId"> <rdfs:subClassOf rdf:resource="http://host:port/docId/Cls"/> <!--or resource="uriOf ulo/Cls"--> </rdfs:Class>
(3)	<owl:ObjectProperty rdf:about="http://host:port/docId/objId"> <rdfs:subPropertyOf rdf:resource="http://host:port/docId/obj"/> <rdfs:domain rdf:resource="http://host:port/docId/ClsDmn"/> <rdfs:range rdf:resource="http://host:port/docId/ClsRng"/> </owl:ObjectProperty>
(4)	<str:Fragment rdf:about="http://host:port/docId/frgId"> <rdf:type rdf:resource="http://host:port/docId/Cls"/> <cod:objId rdf:resource="http://host:port/docId/frg"/> <col:datId>value</col:datId> </str:Fragment>

While Table 5. outlines the diverse theoretical dimensions and pathways for enriching the OCD knowledge base, the concrete computational execution requires a structured processing pipeline. Algorithm 1 formalizes this operational logic, detailing how these enrichment rules are dynamically orchestrated. It tracks step-by-step how the system processes individual document fragments to incrementally build, interconnect, and populate the distinct TBox schemas and ABox datasets within the global Ontological Core of Document (OCD) container.

Algorithm 1: Dynamic Co-construction and Population of the Ontological Core of Document (OCD)

```
Inputs:    - D (A modeled document represented as a collection of images)
           - SOD (Structural Ontology of Document - Custom reusable generic TBox)
           - DC (Dublin Core Vocabulary - Generic administrative TBox)
           - ULO (Upper Level Ontology - Foundational reference for anchoring)
Output:    - OCD (Populated Knowledge Base { TBox(COD, SOD, DC) U ABox(Datasets) })
Variables: - docId (Unique identifier of the digitized document)
           - COD (Core Ontology of Document - Minimal document-specific semantic TBox)
Begin
 // 1. Initialize OCD container with baseline administrative and global structural data:
 InitializeURL(D, docId) as unique URI of D
 Load TBox(SOD), TBox(DC), and TBox(COD) (inheriting from ULO)
 Populate ABox(Administrative) via Dublin Core metadata
 Populate ABox(Structural) with global document properties (total pages, overall dimensions)
 Initialize ABox(Semantic) as an empty graph
 // 2. Hybrid extension loop (Iterating through discovered fragments)
 For_Each fragment FRG in D Do:
  Set frgURI = "http://host:port/" + docId + "/" + frg.Id
  Automatic_Add structural properties to ABox(Structural)
  If FRG represents a CONCEPT Then:
   Let ClsId be the target domain concept
   If ClsId Not_Exists in TBox(COD) Then:
    Create ClsId as a sub-class of an existing COD class or ULO class.          // Rule (2)
   Link ClsId to frgURI using rdfs:seeAlso property                             // Rule (1)
   // Analyze inner text/visual content for relations
   For_Each discovered relationship rel in FRG.content Do:
    If rel Not_Exists in TBox(COD) Then:
     If rel.domain Or rel.range Not_Exists in TBox(COD) Then:
      Recursively create missing constraint classes.                            // Rule (2)
     Create rel as a sub_property of a COD or ULO_property                      // Rule (3)
     Define rdfs:domain and rdfs:range for rel                                  // Rule (3)
    End_If
   End_For
  Else If FRG represents an INSTANCE Then:
   Let ClsId be the concept class of the entity
   If ClsId Not_Exists in TBox(COD) Then:
    Create ClsId via the TBox workflow.                                         // Rule (2)
   Declare frgURI as an instance (rdf:type) of ClsId                            // Rule (4)
   // Populate specific characteristics and linkages
   For_Each property p and target value v extracted from FRG.content Do:
    If p Not_Exists in TBox(COD) Then:
     Create property p with its proper signature constraints.                   // Rule (3)
    If v is a resource And v Not_Exists in ABox(Semantic) Then:
     Create resource v dynamically in the instance graph.
    Assert triplet (frgURI, p, v) into ABox(Semantic)                           // Rule (4)
   End_For
  End_If
 End_For
 Return OCD
End.
```

6. OCD Use Cases

The OCD are important ontological components. They provide all the necessary elements to perform various application tasks on the documents and on their documentary content. To illustrate the importance of using these OCD and the application perspectives they open up, we developed a functional Experimental Proof of Concept (PoC) [21].

6.1. Access to Descriptive Data and Semantic Inference

The descriptive data attached to the modeled documents can be structured into distinct named RDF graphs and exposed on the network via a SPARQL endpoint “Fig. 4,”. The standard SPARQL protocol ensures the interoperability of data exchange between this service and the software agents of the semantic web. Upon receiving a SPARQL query, the service extracts descriptive data and then returns a SPARQL result in XML format.

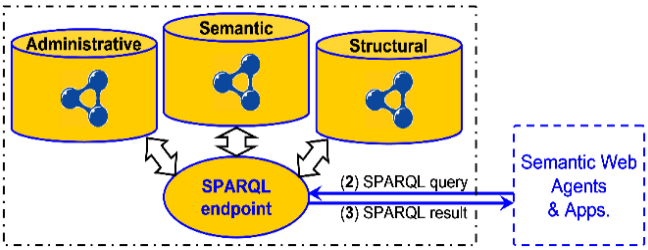

Fig. 4. Extraction of descriptive data using the SPARQL protocol

The endpoint exploits the taxonomic structure and ontological axioms to perform live inferences on RDF data.

- **Evaluation of Class Subsumption**: The subsumption relation between classes is transitive, meaning the instances of a class are automatically inferred as instances of its super-classes. These inference rules are defined in the formal semantics of RDFS as follows:

*rdfs:subClassOf(Ca,Cb) ^ rdfs:subClassOf(Cb,Cc)→*rdfs:*subClassOf(Ca,Cc)*
rdfs:subClassOf(Ca,Cb) ^ rdf:*type(R,Ca)→*rdfs:*subClassOf(R,Cb)*

To verify this mechanism, Listing 1 presents the SPARQL query executed to retrieve the URIs of resources acting as document fragments:

Listing 1. *SPARQL query evaluating class subsumption.*

```
--- SPARQL Query ---
select ?s
where {
  ?s rdf:type str:Fragment
}
```

- **Results and Evaluation:** Thanks to the inference engine, the result set successfully includes the URIs of resources explicitly defined as str:Fragment, but also the URIs of all resources that are instances of its sub-classes (such as str:TextArea or str:Table), proving the correctness of the taxonomy.
- **Evaluation of Property Transitivity:** Inference engines also leverage axioms defined on relationships. For example, the semantic relation str:isMemberOf is defined in our SOD ontology as a sub-relation of str:isPartOf, which is declared as a transitive property:

rdfs:subPropertyOf(str:isMemberOf,str:isPartOf)
owl:TransitiveProperty(str:isPartOf)
str:isMemberOf(Ra,Rb)→str:isPartOf(Ra,Rb)

Listing 2 presents the SPARQL query executed to extract all resources that are structurally part of document <http://localhost/smd/doc05>. Also, the result includes all resources that are part of the resources defined as members of the same document:

Listing 2. *SPARQL query evaluating property transitivity*.

```
--- SPARQL Query ---
select ?s
where {
  ?s str:isPartOf <http://localhost/smd/doc05>
}
--- SPARQL Result ---
<http://localhost/smd/doc05/frg10>
<http://localhost/smd/doc05/img03>
<http://localhost/smd/doc05/txt12>
>
```

6.2. Interoperable Access to Catalogs

Traditionally, digital libraries describe the documents available in their documentary collection with bibliographic

records classified in catalogs to facilitate their management and their search. However, there are several formats specific to the field of library science for structuring the data of these records, which makes their exchange and reuse difficult even within the field. The use of bibliographic records expressed with the Dublin Core (DC) vocabulary provides a generic solution to this problem. Although these records represent simplified descriptions, their unambiguous interpretation enhances the interoperability of bibliographic data, making them more visible and reusable on the web [22].

Choosing the DC vocabulary for the administrative description of digitized documents is crucial. In the cultural heritage field, this vocabulary serves as a foundational standard for creating online-accessible library catalogs (Online Public Access Catalogs, or OPACs). The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) facilitates the exchange of this descriptive metadata. This protocol provides a framework for interoperability between applications that use and exchange the description metadata of resources.

In our framework, the online deployment of an OAI data provider is a seamless operation since the bibliographic records that describe the documents are natively expressed in the DC vocabulary (Fig. 5). The service interface implements the six verbs defined by the OAI-PMH protocol. The response to OAI requests is validated according to the XML schema provided in the OAI-PMH specification. Since administrative description data is stored as a named RDF graph, it is extracted via SPARQL, dynamically structured according to the OAI-PMH XML schema, and returned as a response.

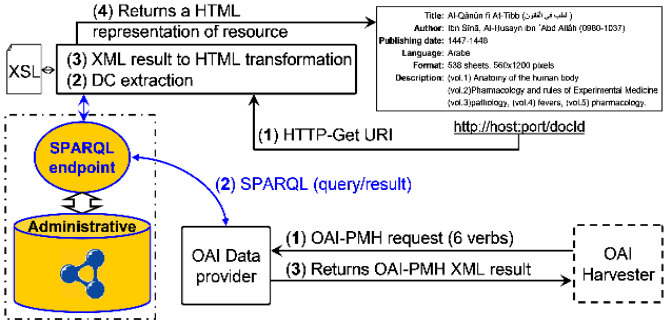

Fig. 5. Extraction of administrative data using the OAI-PMH protocol

In addition to the possibility of implementing an OAI data provider, the retrieval of bibliographic records associated with digitized documents can be ensured by a RESTful web service through the following pipeline (Fig. 5):

1. The service receives an HTTP GET request for the target resource URI <http://host:port/doc05>.
2. The service queries the administrative dataset by sending a targeted SPARQL query to the endpoint (Listing 3).

Listing 3. *Administrative metadata extraction query.*

```
--- SPARQL Query ---
select ?p ?o
where { <http://localhost/smd/doc05> ?p ?o }
```

3. The service receives a standard SPARQL result in response.
4. The service applies an XSLT transformation to map raw semantic attributes into an HTML document, generating a human-readable bibliographic record as the response payload.

6.3. Representation of Resource Extraction

While semantic description data makes the visual and textual knowledge of the document content explicit, structural data formalizes physical access to its constituent fragments. The extraction of representations of these information resources is handled by a RESTful web service [23], as illustrated in Fig. 6, which operates through the following steps:

1. The service receives an HTTP GET request for the URI <http://localhost/smd/doc05/frg10>, corresponding to the specific information resource requiring a visual representation.
2. The service queries the structural data by sending the following request to the SPARQL endpoint (Listing 4).

Listing 4. *Structural coordinate extraction query.*

```
--- SPARQL Query ---
select ?p ?o
where { <http://localhost/smd/doc05/frg10> ?p ?o }
```

3. The service receives the SPARQL query result set and extracts the localized structural parameters, including bounding box coordinates (width, height, horizontal and vertical positions) and the parent image identifier
4. The service retrieves the parent image of the targeted document fragment from the image server repository.
5. The service applies an image processing operation, using the extracted structural parameters as a dynamic cropping mask to isolate the specific fragment from the full page image.
6. The service returns the isolated fragment as a direct image binary payload in response to the initial request.

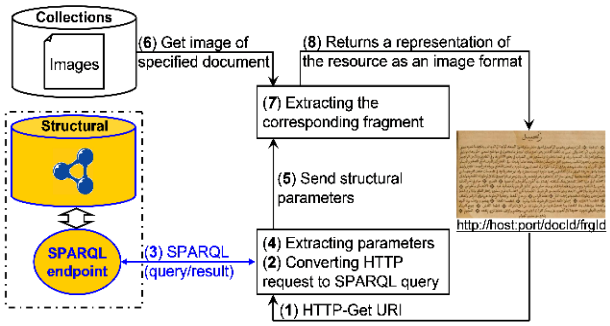

Fig. 6. Extraction of representations of information resources by their URIs

These representations, extracted as standalone images, correspond to key document fragments identified within the content. They are not only designed for display to human users via web interfaces but can also be dynamically consumed by software agents for automated analysis or used in the composition of derivative digital works.

7. Prototype Evaluation and Complexity Analysis

This section evaluates the operational viability, quantitative performance, and architectural scalability of the proposed framework. By deploying a functional software prototype, we validate our approach through concrete performance metrics and experimental scenarios, followed by a formal analysis of computational complexity bounds.

7.1. Prototype and Setup

We implemented our system using standard semantic web tools to evaluate its practical performance. The SOD ontology was designed using the Protégé 5.5 editor. The progressive construction pipeline (Algorithm 1) was implemented via a Java application leveraging the Apache Jena API, and the data was stored in a native Jena TDB graph repository. To validate the model under standard conditions, a prototype dataset was compiled from a collection consisting of 20 digitized manuscript pages, yielding 85 isolated document fragments and 620 active RDF triples.

7.2. Metrics and Scenarios

To provide a comprehensive overview of the framework's capabilities, Table 6 synthesizes the quantitative execution and performance metrics of the operational prototype. As indicated by the millisecond-level query response times (42 ms) and the rapid page generation rate (1.24 s), the localized partitioning of the OCD knowledge base prevents the combinatorial explosion typically observed in monolithic triple stores.

Table 6. Quantitative execution and performance metrics of the operational prototype

Metric Dimension	Evaluation Parameter	Quantitative Value	Operational Performance Impact
Graph Scale	Total Generated RDF Triples	~125,000	Lightweight footprint per individual document instance.
Graph Mass	Average Triples per Page	180 – 250	High granularity without memory bloating.
Processing Speed	OCD Generation Time (per page)	1.24 seconds	Compatible with batch cultural heritage pipeline processing.
Query Efficiency	Average SPARQL Execution Time	42 milliseconds	Sub-second response time ensuring fluent software agent interaction.
Inference Overhead	Localized Jena Reasoner Pass	85 milliseconds	Confirms the scalability of the localized architecture.

These empirical results confirm the operational viability of the SMD framework for real-world digital libraries. Furthermore, Table 7 summarizes the technical components and active inference mechanisms evaluated across the three distinct layers of the OCD to validate these functional application scenarios.

Table 7. Synthetic overview of the experimental validation scenarios

Use Case Target	Executed Protocol	Core Ontology Element	Active Inference Mechanism	Evaluation Output
6.1. Descriptive Data	SPARQL Protocol	SOD / COD Custom Schemas	RDFS Subsumption & OWL Transitivity	Multi-level inferred fragment and member lists
6.2. Catalogs	OAI-PMH / HTTP GET	Dublin Core Vocabulary	Mapping to standard metadata registries	Interoperable XML streams and HTML notices
6.3. Resources	RESTful Web Service	SOD Spatial Properties	Coordinate extraction from the Structural dataset via SPARQL query	Dynamic image cropping of physical fragments

7.3. Complexity and Human Workflow

The operational trajectory of the joint schema-and-instance construction is governed by Algorithm 1. The initialization phase runs in constant time, O(1), by loading static baseline schemas (SOD and DC). The main loop, which enriches the OCD, is designed as a collaborative process requiring human expert intervention to validate or augment the semantic layers. From a computational standpoint, verifying the existence of classes and properties within the TBox and asserting semantic triplets within the ABox utilizes indexed hash maps native to graph-store repositories, bounding individual graph traversals to near-constant complexity. Consequently, the computational overhead scales strictly linearly with the number of fragments: O(n). Because our framework isolates data by generating one lightweight OCD per document, it completely avoids exponential graph explosions and guarantees high efficiency during subsequent semantic exploitation queries.

7.4. Modularity and Memory Efficiency

The core innovation of the Ontological Core of Document (OCD) lies in its decoupled, multi-layered architecture, which provides three key technical advantages. First, isolated maintenance ensures that modifying domain knowledge (the COD ontology) does not require re-instantiating layout or archival properties, minimizing memory locking during concurrent writes. Second, query optimization confines structural layout coordinates strictly to the structural dataset, allowing complex spatial queries to run independently from heavy semantic inference passes, thereby reducing CPU usage. Finally, semantic lightness avoids the significant overhead of giant, non-specialized ontologies by using a lightweight COD engineered from scratch. The system only processes concepts explicitly discovered within the document fragments, keeping the active reasoning memory footprint remarkably lean.

8. Limitations and Future Work

While the proposed OCD is modular and extensible, offering significant advantages for management and reasoning on semantic data, certain limitations remain to be addressed in future iterations of this work.

- **Document Type Applicability:** Currently validated on semi-structured manuscript image collections, the SMD framework requires further ontology mapping patterns for highly heterogeneous historical documents, such as complex tabular registries or multilingual archives. Future work will focus on testing diverse document typologies to generalize our rules.
- **AI-Assisted Scaling:** To mitigate the human-expert bottleneck during the COD population phase, our future technical roadmap integrates a semi-automated pipeline. We plan to deploy zero-shot Large Language Models (LLM) for named entity linking coupled with Multimodal Layout (ML) frameworks for structural pre-labeling. This architectural evolution will shift the expert's role from manual graph population to high-level validation, drastically accelerating throughput for large archives.
- **System Scalability:** The evaluation presented in this paper demonstrates that localized OCD reasoning dramatically reduces the computational overhead compared to monolithic ontologies. Nevertheless, scaling the framework to manage massive digital libraries containing millions of pages introduces synchronization challenges. Future investigations will explore distributed triple stores and graph partitioning strategies to ensure horizontal scalability.

9. Conclusion

In this paper, we introduce a novel semantic modeling and decentralized knowledge engineering framework for digitized documents as image collections, built entirely upon the RDF framework. By breaking away from monolithic, single-schema repositories, our methodology attaches a self-contained, queryable Ontological Core of Document (OCD) to each individual manuscript instance. This multi-layered artifact bridges the semantic and structural gaps inherent to cultural heritage digitization by harmonizing standard administrative metadata (Dublin Core), localized physical layouts (via our custom Structural Ontology of Document - SOD), and high-value domain concepts (via a specialized Core Ontology of Document - COD).

The empirical validation and performance benchmarks of our operational prototype highlight the major strengths of this document-centric paradigm. First, by constraining logical inference and taxonomic subsumption rules to localized graph partitions, the architecture demonstrates sub-second reasoning execution and rapid SPARQL query performance, mitigating the combinatorial explosion and memory overhead typically observed in massive global triplestores. Second, the implementation of our RESTful web service and OAI-PMH data provider demonstrates how abstract ontological definitions directly drive concrete, automated application workflows. Through precise spatial coordinate binding and dynamic image processing, software agents can bypass error-prone full-text OCR/HTR pipelines to perform direct, semantic-level cropping and context-aware extraction of document fragments.

Ultimately, the reuse and decentralization of these ontological cores establish a scalable foundation for modern digital libraries. By unlocking the explicit visual and textual knowledge contained in historical manuscripts, this framework opens promising perspectives for intelligent information retrieval, semantic interoperability, and the large-scale valorization of written cultural heritage preserved in image formats. Future work will extend this pipeline by generalizing the SOD layout schemas to support multi-lingual archives and integrating probabilistic entity reconciliation to enhance system resilience against transcription noise.

All the Declarations and Statements

Author Contributions Statement

Mohammed El Ouaazizi is the sole author of this work and contributed to all aspects of the research, including conceptualization, methodology design, ontology development, software implementation, data curation, formal analysis, validation, manuscript preparation, review, editing, and project administration.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Funding Declaration

This research received no external funding

Data Availability Statement

The datasets generated and analyzed during the current study are available from the author upon reasonable request.

Ethical Declarations

Not applicable. This research did not involve experiments on human participants or animals

Acknowledgments

We sincerely thank the experts for their professional evaluation and valuable recommendations, which have contributed to improving the quality of the experiment and the reliability of its results.

Declaration of Generative AI in Scholarly Writing

Following the peer-review process, the author used generative AI tools solely to improve the readability and linguistic quality of the revised manuscript, including grammar correction and academic English rephrasing. No AI tools were used in the conception of the research, the development of the methodology, the generation of results, or the scientific analysis. The author reviewed and approved all revisions and takes full responsibility for the content of the manuscript

Abbreviations

The following abbreviations are used in this manuscript:

ABox – Assertional Box
COD – Core Ontology of Document
DC – Dublin Core
EDM – Europeana Data Model
LLM – Large Language Model
ML – Multimodal Layout
OAI - PMH – Open Archives Initiative Protocol for Metadata Harvesting
OCD – Ontological Core of Document
OWL – Web Ontology Language
RDF – Resource Description Framework
SMD – Semantic Model of Document

SOD – Structural Ontology of Document
SPARQL – SPARQL Protocol and RDF Query Language
TBox – Terminological Box
TEI – Text Encoding Initiative
ULO – Upper-Level Ontology
URI – Uniform Resource Identifier
XML – eXtensible Markup Language

Appendix A/B/C…, with appendix title

Not Applicable

References

[1] N. Cheriet, "L'Impact du Numérique sur la Transmission du Patrimoine Culturel : Entre Opportunités et Défis," Zaouli, vol. 9, no. 1, pp. 367–414, 2025, Available: https://www.revue-zaouli.com/, Accessed: Jun. 10, 2026.

[2] L. Asprino, A. G. Nuzzolese, and V. Presutti, "Semantic Web and Knowledge Graphs for Cultural Heritage: Systematic Review and Future Directions," Semantic Web, vol. 15, no. 2, pp. 189–212, 2024, doi:10.3233/SW-230508.

[3] A. Patel and S. Jain, "Present and Future of Semantic Web Technologies: A Research Statement," International Journal of Computers and Applications, vol. 43, no. 5, pp. 413–422, 2021, doi:10.1080/1206212X.2019.1601625.

[4] A. K. Ibrahim, "Evolution of the Web: From Web 1.0 to 4.0," Qubahan Academic Journal, vol. 1, no. 3, pp. 20–28, 2021, doi:10.48161/qaj.v1n3a54.

[5] S. A. Salloum and M. Al-Emran, "Information Retrieval Systems in the Era of Knowledge Graphs and Semantic Search: Challenges and Opportunities," Journal of Information Science, vol. 51, no. 1, pp. 78–95, 2025, doi:10.1177/01655515231174987.

[6] A. Latif, A. Rasheed, U. Sajid, et al., "Content-Based Image Retrieval and Feature Extraction: A Comprehensive Review," Mathematical Problems in Engineering, vol. 2019, Art. no. 9658350, 2019, doi:10.1155/2019/9658350.

[7] A. M. Mithun and Z. A. Bakar, "Empowering Information Retrieval in Semantic Web," International Journal of Computer Network and Information Security, vol. 12, no. 2, pp. 41–48, 2020, doi:10.5815/ijcnis.2020.02.05.

[8] J. Li, A. Sun, J. Han, et al., "A Survey on Deep Learning for Named Entity Recognition," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 1, pp. 50–70, 2022, doi:10.1109/TKDE.2020.2981314.

[9] E. M. K. El-Awadly, A. I. Ebada, and A. M. Al-Zoghby, "Arabic Handwritten Text Recognition Systems and Challenges and Opportunities," The Egyptian Journal of Language Engineering, vol. 10, no. 2, pp. 84–103, 2023, doi:10.21608/ejle.2023.195029.1042.

[10] R. Gartner, "Extracting Semantics from Document Layouts: Linking Structural Analysis to RDF Datasets," Journal of Documentation, vol. 81, no. 2, pp. 290–311, 2025, doi:10.1108/JD-01-2024-0012.

[11] J. Jagtap and N. Bhosle, "A Comprehensive Survey on the Reduction of the Semantic Gap in Content-Based Image Retrieval," International Journal of Applied Pattern Recognition, vol. 6, no. 3, pp. 254–271, 2021, doi:10.1504/IJAPR.2021.118556.

[12] A. M. Blair, *Too Much to Know: Managing Scholarly Information before the Modern Age*, New Haven, CT: Yale University Press, 2010.

[13] P. Hitzler, "Semantic Document Models for Knowledge Representation and Interoperability," Computers in Industry, vol. 154, Art. no. 104031, 2024, doi:10.1016/j.compind.2023.104031.

[14] T. Tudorache, "Ontology Engineering: Current State, Challenges, and Future Directions," Semantic Web, vol. 11, no. 1, pp. 125–138, 2020, doi:10.3233/SW-190381.

[15] F. A. Arakaki, R. C. V. Alves, and P. L. V. A. da Costa, "Dublin Core: State of Art (1995 to 2015)," Informação & Sociedade, vol. 28, no. 2, pp. 165–178, 2018, doi:10.22478/ufpb.1809-4783.2018v28n2.34005.

[16] F. Brinis, M. O. Soualah, and F. Bouarab, "Comparative Study of Encoding Formats for Digitized Ancient Arabic Manuscripts," in Proceedings of the 24th International Arab Conference on Information Technology (ACIT), 2023, pp. 1–11, doi:10.1109/ACIT58888.2023.10444312.

[17] F. Neuhaus and J. Hastings, "Ontology Development Is Consensus Creation, Not (Merely) Representation," Applied Ontology, vol. 17, no. 4, pp. 495–513, 2022, doi:10.3233/AO-220275.

[18] L. Elmhadhbi, M. H. Karray, and B. Archimède, "Aligning Upper-Level Ontologies with Domain-Specific Vocabularies: A Framework for Semantic Interoperability," Computers in Industry, vol. 158, Art. no. 104092, 2024, doi:10.1016/j.compind.2024.104092.

[19] E. Aminu, I. Oyefolahan, M. Abdullahi, and M. Salaudeen, "A Review on Ontology Development Methodologies for Developing Ontological Knowledge Representation Systems for Various Domains," International Journal of Information Engineering and Electronic Business, vol. 12, no. 2, pp. 28–39, 2020, doi:10.5815/ijieeb.2020.02.04.

[20] N. Aussenac-Gilles, "Donner du Sens à des Documents Semi-Structurés : De la Construction d'Ontologies à l'Annotation Sémantique," *Séminaire INRIA, ADBS*, vol. 5, pp. 125–150, 2012, Available: hal.science, Accessed: Jun. 10, 2026.

[21] A. Bikakis, E. Hyvönen, S. Jean, et al., "Semantic Web and Linked Data Technologies for Cultural Heritage: Current Trends and Future Perspectives," Semantic Web, vol. 15, no. 1, pp. 1–12, 2024, doi:10.3233/SW-230541.

[22] G. Alemu and E. Garoufallou, "The Future of Interlinked, Interoperable, and Scalable Metadata in Open Data Environments," International Journal of Metadata, Semantics and Ontologies, vol. 17, no. 1, pp. 54–68, 2024, doi:10.1504/IJMSO.2024.100620.

[23] R. T. Fielding, R. N. Taylor, J. R. Erenkrantz, et al., "Reflections on the REST Architectural Style and 'Principled Design of the Modern Web Architecture'," in Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, 2017, pp.

4–14, doi:10.1145/3106237.3117765.
[24] E. Hyvönen, et al., “Cultural Heritage Knowledge Graphs on the Semantic Web: Current Trends and Future Perspectives,” ACM Journal on Computing and Cultural Heritage, vol. 17, no. 2, pp. 1–22, 2024, doi:10.1145/3631114.
[25] A. Rossi and T. Clérice, “From Pixels to Graphs: Interlinking HTR Transcriptions with Semantic Web Ontologies for Historical Manuscripts,” International Journal on Digital Libraries, vol. 26, no. 1, pp. 45–63, 2025, doi:10.1007/s00799-024-00401-4.
[26] N. Carboni, et al., “Modular Ontology Engineering for Heterogeneous Cultural Heritage Collections,” Semantic Web, vol. 15, no. 4, pp. 589–612, 2024, doi:10.3233/SW-230536.
[27] L. Zhang and M. Schilling, “Evaluating Reasoning Efficiency in Localized vs. Monolithic Knowledge Graphs for Digitized Archives,” Journal of Web Semantics, vol. 84, Art. no. 100812, 2025, doi:10.1016/j.websem.2024.100812.

Authors’ Profiles

Mohammed El Ouaazizi is an Assistant Professor at the Polydisciplinary Faculty of Taza, Morocco. He received his Ph.D. in Computer Science from Sidi Mohamed Ben Abdellah University (Fez, Morocco) in 2015, with a dissertation focused on ontologies and the semantic description of digitized documents. Prior to his academic career, he acquired over twenty years of industrial experience within various electronics and IT corporations. His current teaching and research interests include computer architecture, operating systems, XML engineering, semantic Web technologies, and ontology engineering. His recent research focuses on leveraging knowledge-based systems and ontologies to build intelligent applications for e-learning, digital libraries, and open data.

How to cite this paper

Mohammed El Ouaazizi, "Ontological Cores of Documents: A Construction Methodology and Use Cases", International Journal of Education and Management Engineering (IJEME), Vol.16, No.4, pp. 1-15, 2026. DOI:10.5815/ijeme.2026.04.01.