D3.5 DE4A Semantic Toolkit – Initial version

Document Identification			
Status	Final	**Due Date**	31/12/2020
Version	1.0	**Submission Date**	22/03/2021

Related WP	WP3	**Document Reference**	D3.5
Related Deliverable(s)	D3.1, D3.3	**Dissemination Level (*)**	PU
Lead Participant	IHU	**Lead Author**	C. Berberidis
Contributors	T. Karunaratne (SU); Ana Rosa Guzman (MTFP-SGAD); E. Kontopoulos (SU); Syed Iftikhar H. Shah (IHU); I. Konstantinidis (IHU); E. Kapantai (IHU); I. Magnisalis (IHU); M. Losej (SIMPA); Peter Bruhn Andersen (DIGST)	**Reviewers**	H. Metselaar (ICTU/MINBZK) Andreas Berg (eGovlab, SU, SE)

Keywords
semantic toolkit, information desk, information exchange model, canonical evidence types

Document Information

List of Contributors	
Name	Partner
Christos Berberidis	IHU
Syed Iftikhar H. Shah	
Ioannis Konstantinidis	
Eleni Kapantai	
Ioannis Magnisalis	
Thashmee Karunaratne	SU
Efstratios Kontopoulos	
Ana-Rosa Guzman	MTPF-SGAD
Miro Losej	SI-MPA
Peter Bruhn Andersen	DIGST

Document History			
Version	Date	Change editors	Changes
0.1	15/12/2020	IHU	Initial version of document
0.2	15/01/2021	IHU, SU	Chapter 2 – Full text (IHU), Chapter 4 – Revised Structure (SU)
0.3	22/01/2021	SU	Added content in Chapter 4
	22/01/2021	IHU	Added comparison of ontology editors and Triplestore comparison
0.4	28/01/2021	SU	Addition and revision of text in Chapter 4
	29/01/2021	IHU	Added text about XML tools in Chapter 2
0.5	02/02/2021	IHU, SU	Added language's brief introduction (e.g., XML, JSON, OWL/RDF)
0.6	05/02/2021	SU, MTPF-SGAD	Added Chapter 5, modified Chapter 4 content
	05/02/2021	SU	Added section 3.2 in Chapter 3
0.7	10/02/2021	IHU, SU	Added executive summary, introduction and conclusion. Modifications in Chapters 4 and 5
0.8	11/02/2021	IHU, SI-MPA	Added section 3.3 in Chapter 3
0.9	17/02/2021	IHU, SU, DIGST	Added section 3.1 in Chapter 3. Added introduction in Chapter 3. Added Annex I. Final modifications and formatting fixes. Ready for internal review
0.91	03/03/2021	IHU	Minor changes following H. Metselaar (ICTU/MINBZK) (reviewer) comments
0.92	04/03/2021	IHU	Minor changes following Andreas Berg (reviewer) comments
0.93	12/03/2021	SU, MTPF-SGAD	Minor changes in Chapters 4 and 5
1.0	15/03/2021	Atos	Final quality check
1.1	18/03/2021	IHU	Minor changes following Atos quality check
1.2	22/03/2021	Atos	Final version for submission

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable leader	Christos Berberidis (IHU)	12/03/2021
Quality manager	Julia Wells (ATOS)	22/03/2021
Project Coordinator	Ana Piñuela Marcos (ATOS)	22/03/2021

Table of Contents

List of Tables

List of Figures

List of Acronyms

Abbreviation / Acronym	Description
ADMS	Asset Description Metadata Schema
API	Application Programming Interface
BB	Building Block
BBXL	Business Document Metadata Service Location
BPMN	Business Process Model and Notation
BRIS	Business Registers Interconnection System
CA	Consortium Agreement
CCCEV	Core Criteria and Core Evidence Vocabulary
CEF	Connecting Europe Facility
CFS	Certificate on the Financial Statements
CPOV	Core Public Organization Vocabulary
CPSV	Core Public Services Vocabulary
CPSV-AP	Core Public Services Vocabulary - Application Profile
DBA	Doing Business Abroad – one of the three DE4A pilots
DC	Data Consumer
DCAT	Data Catalogue Vocabulary
DE	Data Evaluator
DE4A	Digital Europe for All
DO	Data Owner
DP	Data Provider
DR	Data Requestor
DSD	Data Services Directory
DT	Data Transferor
Dx.y	Deliverable number y, belonging to WP number x
EC	European Commission
EDCI	Europass Digital Credentials Infrastructure
EDM	Exchange Data Model
eID	Electronic Identification
FOAF	Friend of a Friend
IHU	International Hellenic University
IOP	Interoperability
IMI	Internal Market Information system
ISA2	Interoperability solutions for public administrations, businesses and citizens
JSON	JavaScript Object Notation
KG	Knowledge Graph
MA	Moving Abroad – one of the three DE4A pilots
ML	Machine Learning
OOP	Once-Only Principle
OWL	Web Ontology Language
PKI	Public Key Infrastructure

Abbreviation / Acronym	Description
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SKOS	Simple Knowledge Organization System
SPARQL	SPARQL Protocol and RDF Query Language
TL	Task Leader
U	User
UC	Use Case
UML	Unified Modelling Language
UI	User Interface
W3C	World Wide Web Consortium
WP	Work Package
WPL	Work Package Leader
XML	Extensible Markup Language

Glossary

Term	Explanation
Application Profile	An application profile (AP), as yet another group of assets within the 'models' category, describes how a standard is to be applied in a particular domain or application. Standards typically do not contain constraints such as cardinality; these constraints are defined in the application profile. An application profile only applies to the specified domain.
Controlled Vocabulary	Controlled vocabularies are a source of authoritative terms to be entered for values of certain elements, such as personal, family, or corporate names, subjects, and coverage elements.
Criteria	Procedural requirements as conditions to be met and used as a basis for making judgements or decisions in the procedure.
Data Model	A data model contains inherent specifications regarding attribute-level constraints, cross-table relationships, and cardinality.
Knowledge Graph	A semantic knowledge representation formalism that encompasses the schema (i.e., ontology) as well as the actual data (i.e., instances of the concepts in the ontology).
Metadata Standard	A metadata standard is a high-level document which establishes a common way of structuring and understanding data and includes principles and implementation issues for utilizing the standard [27].
Once-only Techical System (OOTS)	OOTS is the technical system currently being built by the European Commission stemming from the Single Digital Gateway Regulation (SDGR) (Article 14 thereof) that will make the principle of once only a reality for key public services, businesses, and citizens across the EU. By December 2023 OOTS will be able to simplify access to cross-border administrative procedures initiated online by citizens or companies based in another EU country [28].
Ontology	An ontology – within the scope of computer and information sciences – can be defined as a formal specification for the purpose of delimiting and grouping instances/concepts (facts, events, entities, elements, etc.), based on their common class (types, properties, interrelationships, etc.), and thus formalizing a full or a subset of a domain [26]. An ontology describes the types of things that exist (classes), the relationships between them (properties) and the logical ways those classes and properties can be used together (axioms) [25].
Scenario	One typical way in which a system is used or in which a user carries out some activity.
Semantic Asset	A specific type of standard which involves highly reusable metadata (e.g., XML Schema, generic data models) and/or reference data (e.g. code lists, taxonomies, dictionaries, vocabularies).
Semantic Component	A component (e.g. Information Desk, Information Exchange Model) of the semantic interoperability framework that uses semantic assets to perform certain functionalities.
Semantic Interoperability Framework	A framework that consists of semantic components and the related semantic assets to facilitate cross-border exchange of evidences.
Taxonomy	A systematic arrangement in groups or categories of concepts according to established criteria
Use case	A specification of one type of interaction with a system. One use case may involve several scenarios (usually a main success scenario and alternative scenarios)
User	Anyone who is a citizen of the EU, a natural person residing in a Member State or a legal person having their registered office in a Member State, and who accesses the

Term	Explanation
	information, the procedures, or the assistance or problem-solving services, referred to in Article 2(2), through the gateway [24].
Vocabulary	A collection of terms for a particular purpose. Vocabularies can range from simple ones, such as the widely used RDF schema, FOAF and DCMI element sets, to complex vocabularies with thousands of terms, such as those used in healthcare to describe symptoms, diseases and treatments. Vocabularies play a very important role in linked data, specifically to help with data integration. For example, metadata vocabulary. The use of this term overlaps with that of 'ontology'.
XML Schema	An XML schema is a description of a type of XML document, typically expressed in terms of constraints on the structure and content of documents of that type, above and beyond the basic syntactical constraints imposed by XML itself.

Executive Summary

This document is an initial version of the implementation of a toolkit for the semantic layer stack in the context of Task 3.3 “Implementation of the semantic tools” for delivering cross-border public services. This implementation process follows an agile methodology, by starting in a baseline level with D3.1 “Initial requirements for semantic assets” and D3.3 “Initial version of the Semantic framework” and incrementally improving by adding the tools resulting from the pilot requirements and the other emerging assets identified and added in these live documents.

To that end, first, the document presents a comparative analysis of the most salient tools and technologies that are available and expected to be used towards the performance of key semantic tasks. The objective was to select the most appropriate set of tools, adjusted to DE4A needs. These tools and the pilot specific ontologies, mentioned in D3.3, provided the basis for the implementation process of the common evidence data models (canonical evidences). Furthermore, the semantic components, introduced in D3.3, for obtaining requisite information and exchanging messages are designed for the first iteration of the project by using a standard approach for ontology modelling with competency questions.

The main results and findings of this deliverable are:

- Based on the tools analysis, it is recommended VocBench as the ontology editor and Virtuoso as the ontology storage tool for the DE4A related needs.
- Identification of the canonical evidence types and implementation of the respective data models in XML Schema format for the first iteration of the project.
- Initial specification and semantic model of the Information Desk (IDK) for helping data evaluators to locate the issuing authorities in the context of the first iteration of the project.
- Initial specification of the DE4A Information Exchange Model (IEM) for the provision of public procedures through modelling the payload of request and response messages for the evidence exchange.

The outcomes of this deliverable will serve as input to WP4 “Cross-border Pilots for Citizens and Business and Evaluation” for the first running phase of the pilots and to WP5 “Common Component Design & Development” for implementing interfaces for authorities involved in the exchange process to make the composition of request and response messages. At the first iteration of the project, the running DE4A pilots will test the properness and validity of the semantic toolkit. Any error or improvement identified during the first iteration will be considered at the next iteration of the project.

This version of the semantic interoperability framework and the tools are lightweight; i.e., they are tailored for the simplest scenarios defined by the DE4A pilots. Major simplifications include that the canonical evidences carry hard coded data that resonates the real examples rather than real data retrieved in real time from live environments. Consequently, a simpler version of the IDK that supports the handling of such mimicked data as well. At this phase, only the ISA^2 Core Vocabularies are being used while more vocabularies and codelists will be considered during the second iteration of the project.

1 Introduction

1.1 Purpose of the Document

The purpose of this document is to describe the initial version of the toolkit of the semantic layer stack for delivering cross-border public services related to the DE4A pilot use cases (Studying Abroad, Doing Business Abroad and Moving Abroad).

Aligned with the determined initial requirements and semantic assets from deliverable D3.1 and based on the initial version of the semantic framework (deliverable D3.3), the initial version of the semantic components and toolkit is presented. The outcome of this deliverable is also in line with the DE4A principles defined in deliverable D2.1 "Architecture Framework", including openness (by using open repositories), effectiveness and efficiency, Once-Only Principle, data minimisation (by using the minimum set of attributes for evidence exchange) and reuse before build (e.g., ISA^2 Core Vocabularies).

The semantic toolkit includes:

- The semantic tools for XML Schema (XSD) management to be used for the implementation of the canonical evidence data models.
- The ontology editors for development of pilots' specific data models in RDF format.
- The RDF triplestores as a type of semantic graph database for storing and managing semantically represented knowledge.
- The implementation of pilots related canonical evidence data models in XSD format that will be part of the payload for evidence exchange.
- The specification of the first iteration of the Information Desk and API implementation that will facilitate DCs and DPs to obtain the required information before making requests and/or sending responses to the respective stakeholders.
- The implementation of the Information Exchange Model for modelling the payload of request and response messages for the evidence exchange.

This deliverable is part of the first iteration of DE4A, therefore, it considers the basic elements that will facilitate the implementation of the Minimum Viable Product (MVP). For this purpose, the canonical evidence types contain the minimum set of common attributes with basic data types (initially based on the ISA^2 Core Vocabularies), cardinalities and codelists. WP3 follows an agile approach. At the first iteration of the project, the running DE4A pilots will test the properness and validity of the semantic components. Any error or suggested improvement identified during the first iteration will be considered at the next iteration of the project.

The outcomes of this deliverable, after collaboration and communication with the related work packages, will serve as input to:

- The Cross-border Pilots for Citizens and Business and Evaluation (WP4) for running the pilot tests for the first iteration with the related Member States that will participate in the pilots.
- The Common Component Design & Development (WP5) for designing the first release of the common components and interfaces that will be used for the running pilots.

1.2 Structure of the Document

The deliverable is structured as follows:

- **Chapter 2** provides an overview and a comparative analysis of the most critical tools and technologies that are expected to be deployed for ontology serialization, and data and information exchange between related stakeholders.

- **Chapter 3** presents the implementation of canonical evidences in XSD format based on data requirements provided as input by the pilots.
- **Chapter 4** describes the semantic model underlying the first iteration of the Information Desk (IDK), along with an initial specification of the corresponding API for accessing information residing in the model.
- In **Chapter 5**, the implementation of the Information Exchange Model (IEM) is analysed focusing on the designing and development process of the model.
- Finally, **Chapter 6** concludes this work and offers future directions of the deliverable.

2 Knowledge Representation & Semantic Modelling: Languages, Tools and Technologies

The recent advancement of the *Semantic Web* considers not only the exchange of data messages amongst systems, but the exchange of knowledge as well. Obtaining high quality information is a challenging task as data should be gathered and filtered from a large, open and frequently changing network of distributed data sources, with blurred semantics and no central control over the data sources' structure. A well-defined and shared common definition and representation of such information exchange could address the problem. Serialization formats like *XML* (*Extensible Markup Language* [1]) and *XSD* (*XML Schema Definition* [2], [3]) provide the means for establishing interoperability in information systems.

Nevertheless, increased needs for more advanced semantics-enabled knowledge representation led to the introduction of ontologies, which define the concepts and relationships for describing and representing a domain of discourse [4]. The key languages for formalizing ontologies are *RDF* (*Resource Description Framework* [5]), *RDF/S* (*RDF Schema* [6]), *OWL* (*Web Ontology Language* [7]), and the more recent *SHACL* (*Shapes Constraint Language [*[8]).

The design, development and management of ontologies is a nontrivial task and requires close cooperation between domain experts and knowledge engineers. There is, thus, a substantial need for sophisticated ontology management tools, like dedicated editors for authoring the ontologies and specialized databases (called *triplestores*[1]) for storing and managing the semantic models.

This chapter offers an overview of the relevant knowledge representation languages (XML, XSD, JSON-LD, RDF, OWL, SHACL), tools (ontology editors) and technologies (triplestores and ontology management frameworks). A thorough comparison between the tools is conducted based on a set of required features for DE4A, and the results along with discussions from the experiments based on the performance of each tool are also presented in the forthcoming sections. Our aim is to select the most potent tools that will be used for implementing the models, vocabularies, and semantic tools presented in the following chapters, based on the requirements specified in deliverables D3.1 and D3.3.

2.1 Information Exchange and Knowledge Representation Languages

This section briefly presents the information exchange language XML and its schema definition language XSD, followed by an overview of ontology languages RDF and OWL and JSON-LD, which is a JSON extension for encoding linked data using JSON.

2.1.1 XML and XSD

XML is a W3C-recommended open standard language providing a software- and hardware-independent way of storing and sharing data. Contrary to HTML, XML has no predefined tags, and authors typically define both the tags and the document structure. Due to its openness and extensibility, XML can be easily processed both by humans and machines.

Moreover, XSD provides a schema definition formalism "on-top" of XML data, by detailing the allowed elements, attributes and data types associated to attributes in an XML document. In this context, an XML document is considered "*well-formed*", if it meets the W3C requirements for being XML, and "*valid*", if it is well-formed and meets the requirements imposed by a specified XSD schema. For more advanced validation capabilities, Schematron[2] can also be considered, which is a rule-based validation language for making assertions about the presence or absence of patterns in XML trees.

[1] https://en.wikipedia.org/wiki/Triplestore

2 https://schematron.com/

2.1.2 RDF and RDF Schema

While XML is primarily a serialization format, RDF (Resource Description Framework) is primarily a data model and encompasses a group of W3C specifications for conceptual description of information. The fundamental elements are resources, which are included in statements in the form of *subject-predicate-object* expressions (e.g. person_1 isChildOf person_2), also identified as *triples*. Predicates are properties of the subject resource, whose values (objects) may be another resource or a literal, such as string or integer. Furthermore, RDF's schema language, RDF Schema (also known as RDFS), is an extension of the basic RDF vocabulary and provides a data-modelling vocabulary for RDF data.

2.1.3 OWL, OWL 2 and SHACL

The Web Ontology Language (OWL) is a powerful language for defining ontologies on the Web and consists of a family of sub-languages that are based on formal semantics. An OWL ontology represents a domain in terms of classes, properties and individuals and typically includes very rich explanations of the attributes of those objects or the constraints governing them. OWL 2 is currently the most recent version of the language.

On the other hand, SHACL is a W3C specification for validating graph-based data against a set of conditions, by defining an RDF vocabulary to describe shapes, i.e., collections of constraints that apply to a set of nodes. Although SHACL's initial focus area was data validation, it has evolved into a vehicle both for describing and constraining the contents of an RDF graph, making it an extremely useful new technology. The SHACL standard is now being increasingly adopted by major industrial players.

2.1.4 JSON and JSON-LD

JSON (JavaScript Object Notation) is a lightweight syntax for storing and exchanging data and shares several characteristics with XML (openness, extensibility), offering at the same time a greater degree of user-friendliness. The JSON format signifies the data in the form of objects, i.e., key-value pairs and JSON's simplicity has rendered it a favoured data exchange format for several languages, utilized especially for asynchronous client-server communication.

JSON-LD is a lightweight JSON based format for representing *Linked Data* [9]; the latter data adhering to an ontology schema. Based on the already effective JSON format, JSON-LD is simple for humans to read and write and offers a way to help JSON data interoperate at Web-scale. JSON-LD is considered very suitable for programming environments, REST Web services, and unstructured databases. JSON-LD also serves as an alternative serialization format for RDF data and RDFS graphs, and, at the same time, being based on JSON, it allows for reuse of existing JSON parsers and libraries, facilitating the seamless integration between modern Web-based programming environments.

2.2 XML Schema Editors

This section presents the most popular XSD editors, which were used for the work presented in the next chapter revolving around the implementation of canonical evidence.

2.2.1 Altova XMLSpy

Altova XMLSpy[3] is a tool for XML and JSON validation and processing, offering value beyond basic validation checking. XMLSpy abstracts the complexity of editing XML through an intuitive user interface (UI) and a rich variety of views and options. Offered features include graphical views, code generators, wizards, and other intelligent JSON and XML editing functionality. The tool also provides intelligent guidance and entry-helpers as users type, along with a fast and easy troubleshooting mechanism.

[3] https://www.altova.com/xmlspy-xml-editor

Additionally, XMLSpy provides an XML Schema editor that permits the user to create schemas in a visual, drag-and-drop manner (see Figure 1), also offering functionalities, like, e.g., (a) generation of XSD from XML, JSON Schema, or relational databases, (b) sample instance generation from XSD, and (c) Java, C#, and C++ code generation based on XML Schema.

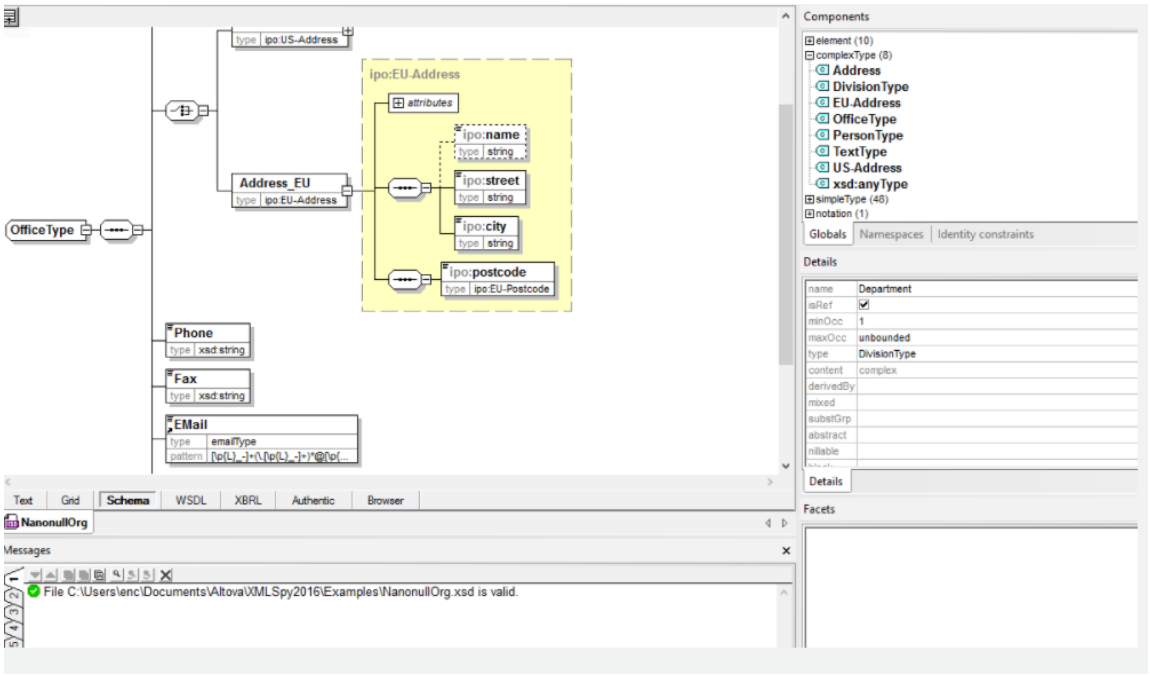


Figure 1: XMLSpy UI.

2.2.2 Sparx Enterprise Architect

Sparx Enterprise Architect is a multi-user, graphical tool that supports technical teams in building robust and maintainable systems (see Figure 2). Users can trace high-level specifications to analysis, design, implementation, test, and maintenance of data models using UML, and BPMN and other open standards. As the tool creators suggest, Enterprise Architect is a spectacularly fast performer, loading extremely large models in seconds[4].

[4] https://sparxsystems.com/products/ea/index.html

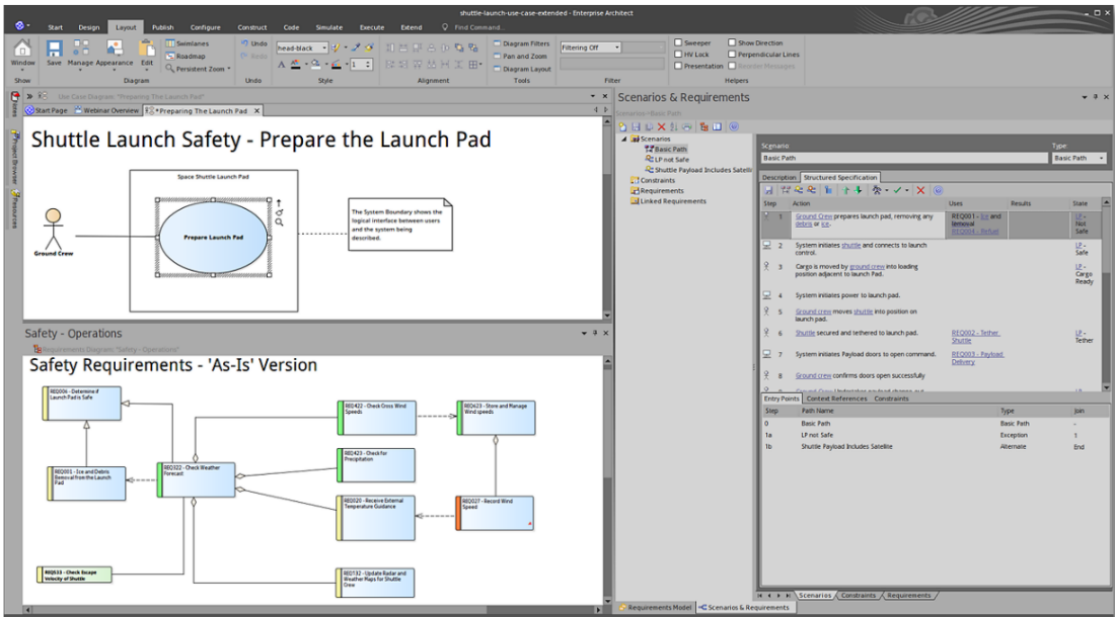


Figure 2: Sparx Enterprise Architect main window.

2.3 Ontology Editors

This section presents the most popular tools for ontology editing, followed by a thorough comparative analysis.

2.3.1 Protégé

The ontology engineering tool Protégé[5] was developed by the Stanford University School of Medicine, USA. It is a free, open-source, and W3C standards-compliant platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. Its initial version was released in 1999, while its last version 5.5.0 was released on 15th March 2019. With a user base of over 350K users, Protégé constitutes a full-fledged user-friendly graphical and interactive ontology design and knowledge acquisition environment. Its component-based architecture enables third parties to add new functionality by creating appropriate plug-ins.

Protégé has desktop and web-based versions, i.e., Protégé Desktop (see Figure 3) and Web Protégé. Protégé Desktop has various features which are required for ontology editing with full support for the OWL 2 Web Ontology Language, and direct in-memory connections to Description Logic reasoners such as HermiT and Pellet. It supports the creation and editing of one or more ontologies in a single workspace through a completely customizable user interface. It also contains visualization tools, which allow for interactive navigation of ontology relationships, advanced clarification support that is helpful in tracking down inconsistencies, refactor operations including ontology merging, moving axioms between ontologies, renaming of multiple entities, and more.

[5] https://protege.stanford.edu/

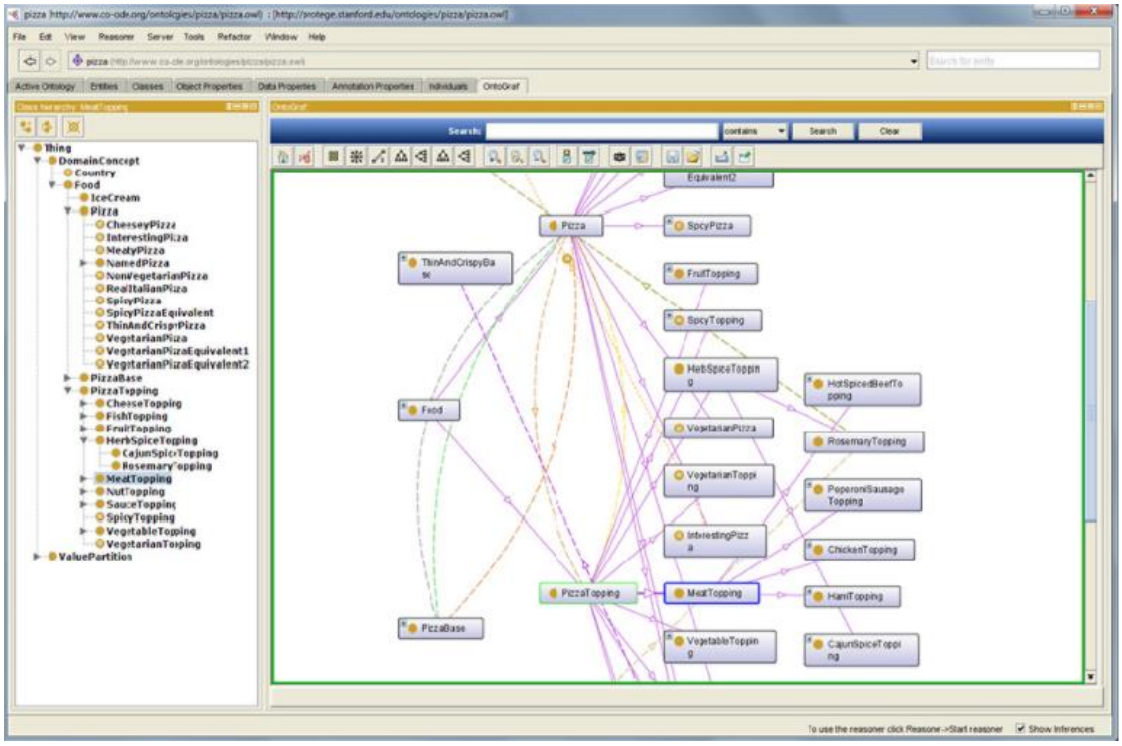


Figure 3: Protege desktop application UI.

The web version of Protégé provides a friendly and highly configurable ontology development environment for the Web, making it easy to create, upload, modify, and share ontologies for collaborative viewing and editing. Collaboration features abound, including sharing and permissions, threaded notes, and discussions, watches and email notifications. Turtle, OWL/XML, RDF/XML, OBO, and other formats are accessible for ontology upload and download for the users.

2.3.2 VocBench

VocBench is a web-based, multilingual, and collaborative development platform to facilitate editing and management of ontologies, designed to meet the needs of Semantic Web and Linked Data environments. It manages multi-lingual controlled vocabularies like ontologies (OWL 2), thesauri, authority lists, glossaries, lexicons, and generic RDF. It also permits users to maintain, validate and publish content through a flexible group management environment. VocBench was originally released by the Food and Agriculture Organization of the United Nations and the Artificial Intelligence Research Group of the University of Rome Tor Vergata, while a new, completely reengineered version of the system (VocBench3), funded by the ISA[2] programme of the European Commission[6], was released in 2017, broadening the scope of the platform [10], [11].

The current version, VocBench3 (v6.0), has various functionalities that include but are not limited to offering connectors to repositories and registries for the provisioning of datasets, advanced concept management, improved visualisation of graphs and resources, filters and operations on nodes in the graph, improved Sheet2RDF editing with a more powerful wizard, improved user interface of metadata management, and SPARQL support[7], and interaction with external triplestores. It also includes access control and user administration, group management, and Role Based Access Control (RBAC) features that permit flexible roles for maintenance, validation, and quality assurance.

[6] ISA[2], EC, Website Page – VocBench: https://ec.europa.eu/isa2/solutions/vocbench3_en

[7] SPARQL (SPARQL Protocol and RDF Query Language) is the standard query language for RDF [12].

The VocBench community is expanding and it also includes FAO's Fisheries and Aquaculture Department and the data.fao.org project, the European Commission Publications Office and the European Environment Agency. Figure 4 illustrates VocBench's main window.

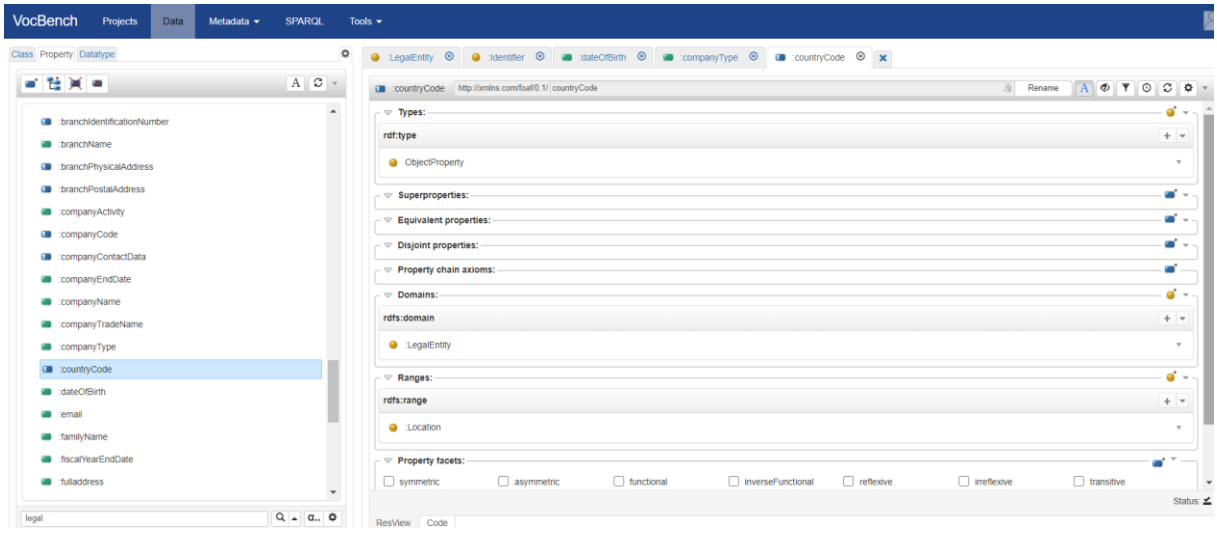


Figure 4: VocBench main window.

2.3.3 TopBraid Composer

TopBraid Composer[8] (TBC) is a modelling interface for developing and managing ontologies and related applications. Being applicable to W3C standards, TBC is a powerful RDF and ontology editor with SPARQL capabilities. The interface is based on the Eclipse platform and the Jena API[9], and can be used to develop data models, and to convert them to or from RDF representation as well as to transform and integrate different data sources.

TBC consists of three editions:

- Free Edition (FE): FE includes all the basic functionalities for the creation and maintenance of ontologies. FE is no longer supported by TopBraid and the latest version is 6.0.1 last modified in September 2018. Figure 5 illustrates the user interface of the FE.
- Standard Edition (SE): SE extends FE by including importing facilities and graphical viewers. SE is no longer supported by TopBraid and the latest version is 6.1.1 last modified in April 2019.
- Maestro Edition (ME): ME extends SE by including Web Application development capabilities and more features. The current version of ME is 6.4.2 last modified in November 2020. ME also includes a 30-day free trial.

Below are some of the key features provided by TBC:

- Autocompletion, drop-down lists and wizard functionalities for easier ontology management and editing.
- Visual editors with diagrams representing classes and instances providing a comprehensible view of the ontologies and data.
- Automation in transforming ontology schemas (RDFS, OWL) to SHACL shapes [8]and in producing SHACL shapes from instances.
- Linking semantic assets or ontologies with geospatial ontologies.
- Importing ontologies or data from files or databases.

[8] https://www.topquadrant.com/products/topbraid-composer/

[9] https://jena.apache.org/

- Multiple data sources integration and transformation: TBC allows transforming RDF to and from XML and XML Schemas, spreadsheets, RDBMSs, JSON.
- Semantic inferencing.
- Ontology matching.
- Automated production of SPARQL queries based on the visual editor.
- Auto-completion capabilities when querying.

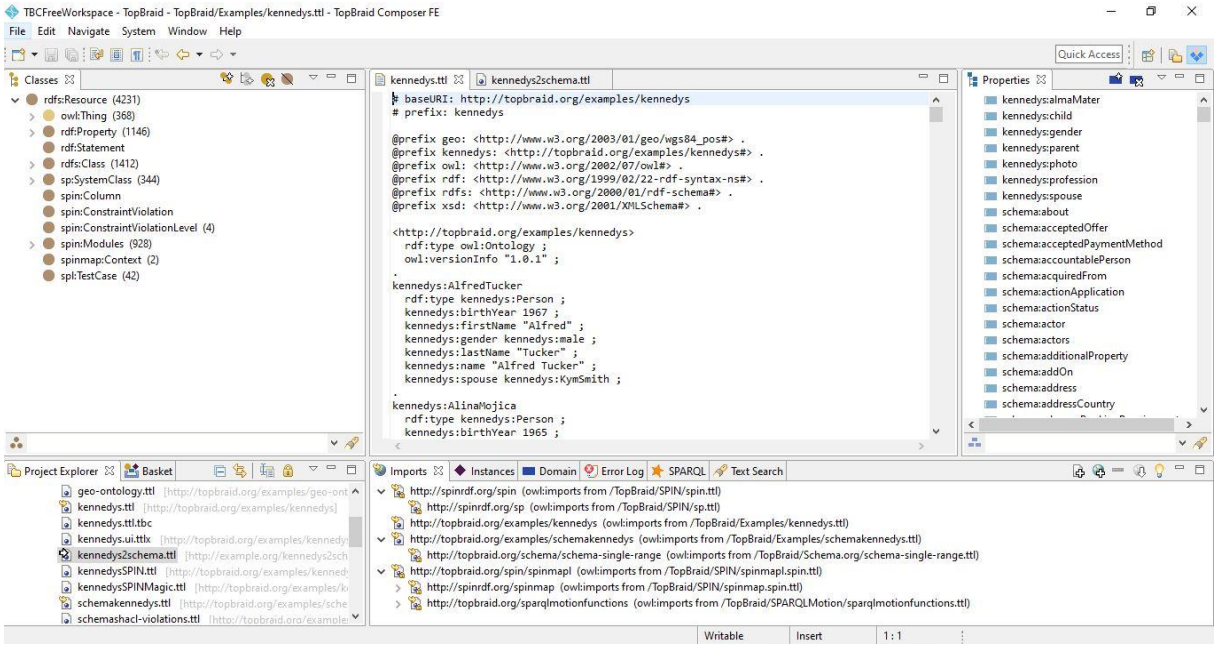


Figure 5: TopBraid Composer FE v6.0.1 UI.

2.3.4 Fluent Editor

Produced by Cognitum, Fluent Editor[10] is an award-winning comprehensive tool for editing, manipulating and querying complex ontologies written in OWL, RDF or SWRL [13] using Controlled English, which is the tool's main and most prominent feature. Controlled English is a subset of the English language with restricted grammar and vocabulary in order to reduce its ambiguity and complexity. Based on the Ontorion Controlled Natural Language (OCNL), the editor provides a user-friendly interface for users who are not familiar with XML principles prohibiting them from entering any sentence that is grammatically or morphologically incorrect, along with a cloud-based scalable solution for storing and managing large ontologies which significantly distinguishes it from other editors.

Fluent Editor is fully compatible with most of the Semantic Web W3C standards (OWL, RDF, RDFS, SPARQL, SKOS) and extremely helpful for beginners in academia and personal use [14]. Key features include:

- Edit ontologies/ Create data models using natural language (OWL syntax knowledge is not needed).
- Export from Controlled Natural Language (CNL) format to OWL format.
- Integrate OWL applications with open source CNL API library.
- User guide editing mode (Hints, Explanations, Predictive Editor, Taxonomy preview).
- Manipulate and query knowledge with Ontorion servers directly.

Furthermore, there are two plugins: a Protégé interoperability plugin (to export/import to/from Protege) and an R plugin that uses the ROntorion pakages to plot and list the current content of the ontology.

[10] http://www.cognitum.eu/semantics/FluentEditor/

2.3.5 Comparison of Ontology Editors

This section conducts a comparison of ontology editors based on key features. Such features include but are not limited to ease of use, editor's openness to other technologies and tools, compliance to standard formats, data handling and collaborative environment options. Standard formats and protocols of the ontology editors in order to satisfy the demands and purposes of this work in terms of the semantic toolkit development are also considered. The features for each ontology editor are presentedin Table 1.

Table 1: Ontology editors – Key features and comparison.

Feature	Sub-Feature	Importance for DE4A	Protégé	VB	TBC	FE
Tools related	Open Source and free of cost	High	Yes	Yes	Yes (TBC FE)	Yes
	Community Support	Medium	Strong	Medium	Strong	Medium
	Initial version	Low	Nov, 1999	2008	May, 2006	Mar, 2011 (beta)
	Launched by	Low	Stanford University, US	ISA_2 EU	TopQuadrant	Cognitum Company
Compliance to standard formats, interface compliance, formats and protocols and API	RDF	High	Yes	Yes	Yes	Yes
	OWL	High	Yes	Yes	Yes	Yes
	SKOS	High	Yes	Yes	Yes	Yes
	SHACL	High	Yes	Yes	Yes	Yes
	XML	High	Yes	Yes	Yes	Yes
Installation complexity		Medium	Low	Low	Low	Low
Collaboration environment support		High	Yes	Yes	Yes	Yes
Usability	Ease of use	High	High	High	High	High
	Open access to updated user documentation and tutorials	Medium	Yes	Yes	Yes	Yes
	Multi-lingual interface	High	No	Yes	Yes	No
Level of Integration with other open-source tools and plug-ins		Medium	Excellent	Mediocre	Mediocre	Low
Tool Extendibility (in terms of additional functionalities)		low	Yes	Yes	Yes	Yes
Import and export of concept models (UML)		Medium	Yes	Yes	Yes	Yes
Built-in interface for SPARQL		High	Yes	Yes	Yes	Yes

Feature	Sub-Feature	Importance for DE4A	Protégé	VB	TBC	FE
View of the concepts (classes, sub-classes, object property, data type property) within interface		Medium	Yes	Yes, limited options	Yes	Yes
Graph-based visualization of concepts and relations		High	Yes	Yes	Yes	Yes
Repository administration	User administration and role-based access	High	No	Yes	Yes	Yes
	Statistics and reporting (Ontology metrics)	High	Yes	Yes	Yes (SPIN statistics for SPARQL rules)	No
	Error warning, tracing, and handling	Medium	Yes	Yes	Yes	Yes
Eease in publishing, exporting, and performing advanced searches about ontologies		High	High	High	Medium	Medium
Technological issues like confidentiality, integrity, accessibility, scalability etc.		Medium	No	No	No	No
User tracing		Low	Yes	Yes	Yes	Yes
Client interface type		Low	Desktop and Online	Online	Desktop and online	Desktop

As observed from the table, ontology editors slightly differ from one another. However, their differences are of minor importance in terms of ontology management and development. It is pertinent to mention that all of the above-mentioned ontology editor tools could be used for the DE4A related needs. However, the VocBench ontology editor is preferred for the DE4A needs, since it as a product from an EU project that also provides multilingual support.

2.4 RDF Triplestores

RDF triplestores are a type of semantic graph database for storing semantically represented knowledge in the form of subject–predicate–object triples based on the Resource Description Framework (RDF), utilizing inference to deducing new information out of the existing facts. Triplestores typically utilize intelligent data management solutions, which blend full text search with graph analytics and logical reasoning to generate deep and semantically rich results. Currently, there exist several high-quality open source and commercial triplestores. This section provides an overview of three key representatives, Graph DB, Virtuoso, and Allegrograph, followed by a feature comparison that justifies our final choice of preferred ontology storage for DE4A.

2.4.1 GraphDB

GraphDB[11] by OntoText is a family of highly efficient, robust, and scalable RDF triplestores, which permit users to link diverse data, index it for semantic search and enrich it via text analysis to create big knowledge graphs. GraphDB offers tools for facilitating the search of semantic knowledge through complex searches, retrieving domain objects to rank results by relevance, finding similar resources with knowledge graph and word embeddings, executing known queries in milliseconds, and building additional data indexes. In order to establish compatibility with the industry standards, GraphDB employs the RDF4J framework interface[12], the W3C SPARQL Protocol, and supports all RDF serialization formats. Furthermore, GraphDB offers a JDBC-like user API, streamlined system APIs and a RESTful HTTP interface. GraphDB is one of the few triplestores that offers semantic inferencing at scale, permitting users to derive new semantic facts from existing facts. It can handle huge loads, queries, and inferencing in real time.

GraphDB offers fast integration of new information sources by parsing structured data in CSV, XLS, JSON, XML or other formats, cleaning the input data with a generic script language, reconciling input datasets to knowledge graphs, generating RDF data and storing it in a local or remote SPARQL endpoint, automating the process and repeat in batch mode.

GraphDB is the favoured choice of both small independent developers and big enterprise firms because of its community and commercial support, and excellent enterprise features like cluster support and integration with external high-performance search applications - Lucene, Solr, and Elasticsearch. Additionally, GraphDB is more stable compared to other popular triplestores as reported in [15].

Ontotext offers the following three editions of GraphDB: GraphDB Enterprise Edition (EE), GraphDB Standard Edition (SE), and GraphDB (free). GraphDB EE and GraphDB SE are both commercial: the former consists of a high-availability cluster with worker and master database implementation for resilience and high-performance parallel query answering, while the latter is file-based, scales to tens of billions of RDF statements on a single server and can handle an unlimited number of concurrent queries. Finally, GraphDB Free is a file-based, scales to tens of billions of RDF statements on a single server with a limit of two concurrent queries. Figure 6 shows the UI of the GraphDB triplestore.

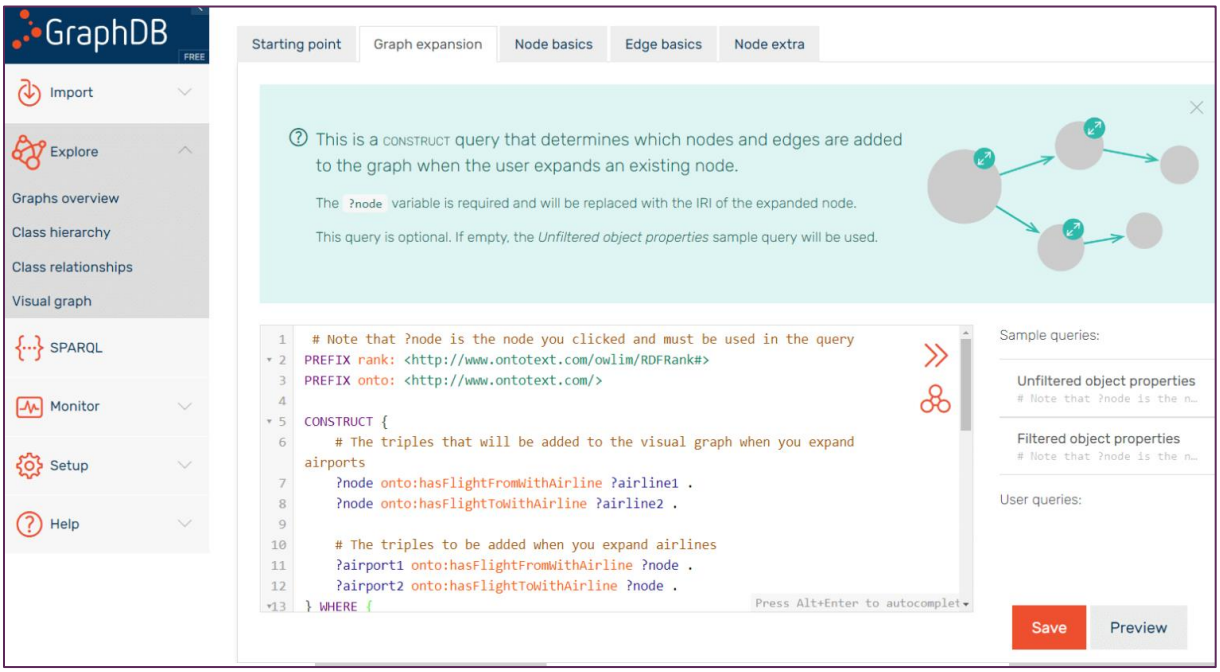


Figure 6: GraphDB main UI.

[11] https://www.ontotext.com/products/graphdb/

[12] https://rdf4j.org/

2.4.2 Virtuoso

Virtuoso was developed by OpenLink Software and was launched in 1998 as an SQL database. It now is a multi-model hybrid-RDBMS that supports the management of data represented as relational tables and/or property graphs (see Figure 7). It is a database tool that integrates various functionalities of multiple systems like Relational Database Systems (RDBMS), object-RDBMS, virtual databases, RDF, XML, plain text, and web application server. Virtuoso is a universal server, meaning that it does not include dedicated servers for the abovementioned systems. Therefore, it allows a single server process with multiple threads and protocols. Virtuoso also includes a free version named OpenLink Virtuoso[13].

Virtuoso allows various ways for querying RDF data:

- SPARQL Query Service Interface (using SPARQL with HTTP connection).
- All SQL Interfaces to Virtuoso (ODBC, JDBC, OLEDB, ADO.NET, and XMLA).
- Virtuoso Stored Procedures and Functions.

Virtuoso is also an OWL reasoner, supporting basic reasoning operations based on owl:sameAs, rdfs:subClassOf, and rdfs:subPropertyOf.

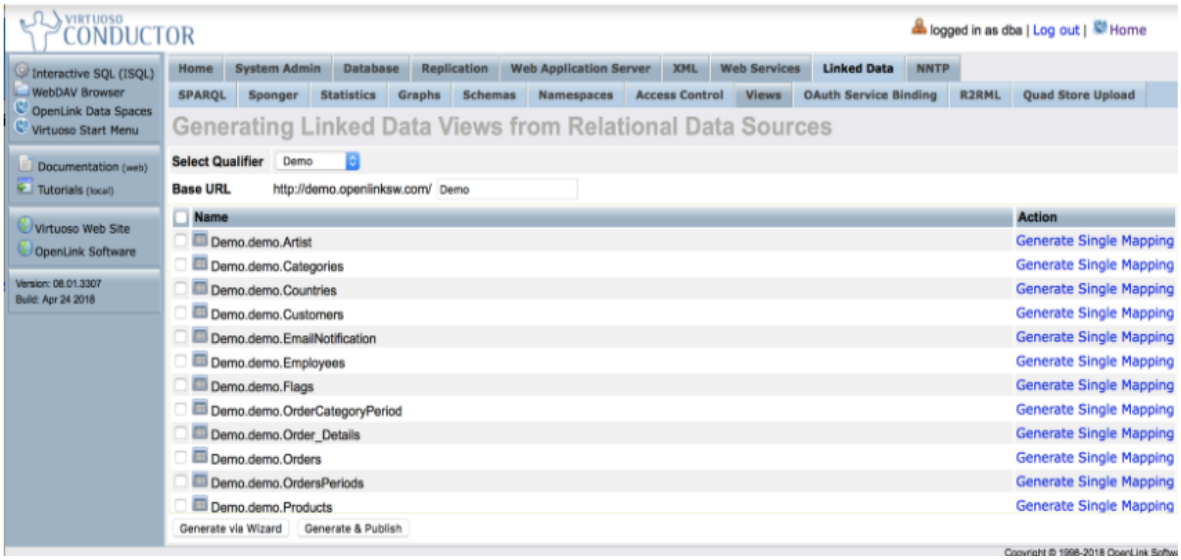


Figure 7: Virtuoso UI view.

2.4.3 AllegroGraph

AllegroGraph (AG) is a semantic graph database focused on generating sophisticated semantic knowledge graphs based on a high performance triplestore. The first version of AllegroGraph was made available at the end of 2004 by Franz Inc. It is an ongoing project with the latest stable release 7.0.4 in November, 2020 (see Figure 8).

As it was developed to meet W3C standards for RDF, it is properly considered an RDF database. However, AllegroGraph doesn't restrict the contents of its triples to pure RDF. In fact, the product is optimized for storing and retrieving any graph data-structure by treating its nodes as subjects and objects, its edges as predicates and creating a triple for every edge. AllegroGraph can load data in the following RDF formats: JSON-LD, N-Quads, N-Triples, Extended N-Quads, RDF/XML, TriG, TriX, Turtle as well as in several non-RDF formats, like JSON, JSONlines, CSV.

Data and metadata can be managed using Java, Python, Pearl, Ruby, C#, HTTP, Javascript, Lisp, Clojure and Scala interfaces, and queried using SPARQL and Prolog. Furthermore, AllegroGraph supports several specialized datatypes for efficient storage, manipulation, and search of Social Network, Geospatial and

[13] https://virtuoso.openlinksw.com/

Temporal information. Finally, AllegroGraph works with various other database tools, some of which organize data stores and others of which provide specialized indexes.

Other capabilities include:

- The product's primary focus is on transactional processing; however, it is often used for analytics as well. Consequently, it is OLTP-enabled (Online Transactional Processing) and fully ACID- compliant (atomicity, consistency, isolation, durability), and additionally offers immediate consistency.
- The product is also highly secure and supports the requirements for various government security standards.
- It includes a wealth of features, including distributed deployment and querying, multi-modal ingestion, multi-master replication, AI and machine learning, and natural language processing (NLP).
- AllegroGraph FedShard™ is the newest feature offering massive horizontal scalability. It enables scaling using multiple repositories on multiple servers. A large dataset is partitioned into shards based on some classifying criterion, and each shard can have access to a common knowledge- base. Queries issued against the distributed repository are run in parallel on each of the shards and afterwards the results are combined. This unique data federation capability allows running highly complex queries across highly distributed datasets and knowledge bases very efficiently.

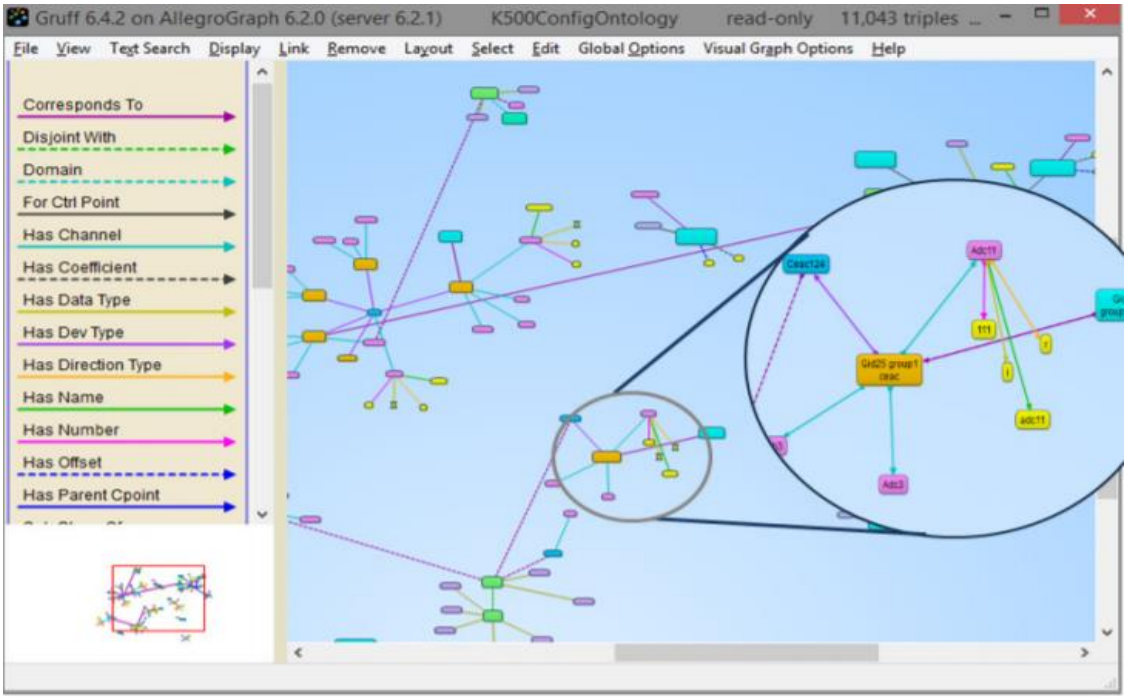


Figure 8: Allegrograph UI for browsing stored ontologies.

2.4.4 Comparison of Ontology Storage tools

This section conducts a comparison analysis of ontology storage tools based on key features that are presented in Table 2.

It is observed that the ontology databases slightly differ from one another, considering as evaluator the importance to DE4A and focusing on "high" labelled features. Our analysis indicates that GraphDB slightly outperforms the comparison and could be preferred to play the role of the RDF triplestore in our semantic toolkit.

Table 2: Ontology management tools – Key features and comparison.

Feature	Importance to DE4A	Tool		
		Graph DB	Virtuoso	AG
User Friendly	High	High	Medium	Medium
Initial release	Low	2010	2006	2004
Current Release/Edition	Low	9.5 edition, September 2020	7.2.5.1 edition, August 2018	7.0.4 edition, November 2020
Development language	Low	C#	C	Java, Python, Common Lisp
Server operating systems	High	Windows, Linux, MAC OS X	Windows, Linux, MAC OS X	Windows, Linux, ,MAC OS X
SPARQL support	High	Yes	Yes	Yes
DBMS Model	Low	Graph DBMS, RDF Store	Document store Graph DBMS Native XML DBMS Relational DBMS RDF store Search engine	Graph DBMS, RDF Store, Document Store
Supported data types	High	Structured, and unstructured data types, all RDF data types	All RDF data types, including language-tagged and XML schema typed strings as native data types	RDF, JSON, XSD
Inference Type Supported	High	RDFS, OWL2RL	RDFS, OWL	RDFS, OWL
Support for quadruple stores	Low	Yes	Yes	Yes
Supports full text search,	Medium	Yes	Yes	Yes
Open source, commercial or both	High	Both	Both	Both
Client interface type (Desktop, Web-based)	Medium	Web-based	Web-based	Both *(AllegroGraph Web View (AGWebView) is a browser based graphical user interface for exploring, querying, and managing AllegroGraph databases.)*

Feature	Importance to DE4A	Tool		
		Graph DB	Virtuoso	AG
DB-Engines Ranking [19]	High	Rank: 5 Score: 2.11	Rank: 4 Score: 2.15	Rank: 7 Score: 1.19
Data scheme	High	Schema-free	SQL, RDF-Quad or Triple XML-DTD, XML Schema	RDF schemas
Support for concurrent manipulation of data	High	Yes	Yes	Yes
Stability of the tool	Medium	High	Medium	-Medium

2.5 Conclusions

This chapter presented an overview and a comparative analysis of the most salient tools and technologies that are available and expected to be used and facilitate our work towards the performance of key semantic tasks of this deliverable, such as data exchange and ontology serialization. The selected tools were examined and evaluated based on their features and their importance to the DE4A project. To monitor and highlight major strengths or weaknesses among them, the "importance to DE4A" metric was defined. The goal was to select the most appropriate set of tools, adjusted to DE4A needs. Based on the previous analysis, it is recommended VocBench as the ontology editor and Virtuoso as the ontology storage tool for the DE4A related needs.

3 Implementation of Canonical Evidence

Design and implementation of the canonical evidences is a major task carried out by the DE4A WP3. This chapter contains concise descriptions of the implemented common data models that serves as canonical evidences, as required by the three pilots Studying abroad (SA), Doing business abroad (DBA) and Moving abroad (MA) for cross-border evidence exchange. The data models are constantly improved based on the requirements from the pilot participants as well as taken into account of the maturity of the existing resources, primarily from the ISA2 programme and the Semic SDG-OOP WP4 - Data Semantics, Format and Quality. The status of the canonical evidence descriptions reported in here corresponds to the versions as at 28th of February 2021.

During the first iteration of the piloting of minimum viable product (MVP) in selected Member States, the following canonical evidences are developed within the DE4A semantic interoperability framework and the toolkit.

1. HigherEducationEvidence (SA)
2. CompanyRegistration (DBA)
3. ResidencyProof (MA)
4. BirthEvidence (MA)
5. MarriageEvidence (MA)

The base for the canonical evidences is provided by the ontology descriptions presented in the Deliverable D3.3: Semantic Framework - Initial version, and the pilot deliverables for requirement elicitation, i.e., D4.1: Studying abroad - Use Case Definition & Requirements, D4.5 Doing Business Abroad - Use Case Definition & Requirements and D4.9 Moving Abroad - Use Case Definition & Requirements. The common data models for the canonical evidences are developed in a co-creation setting together with the semantic experts, domain experts and the member state representatives from each of the piloting teams. An agile approach is followed during the build of the models.

3.1 Canonical Evidences development process

The canonical evidences development process consisted of three stages as illustrated in Figure 9, and followed an incremental development methodology under agile principles. Hence, in this setting, the WP leader plays the role of the product owner, working in a technical working group with the other 2 technical works packages, WP4 (Cross-border Pilots for Citizens and Business and Evaluation) and WP5 (Common Component Design & Development), following the same development cycles, as described in the DE4A grant agreement. Therefore the canonical evidences development process includes the users (the piloting member states) since early stages of the process, allowing the creation of the data models together with the user rather than for the user.

The key stages of the process include elicitation, development and evaluation of the data models.

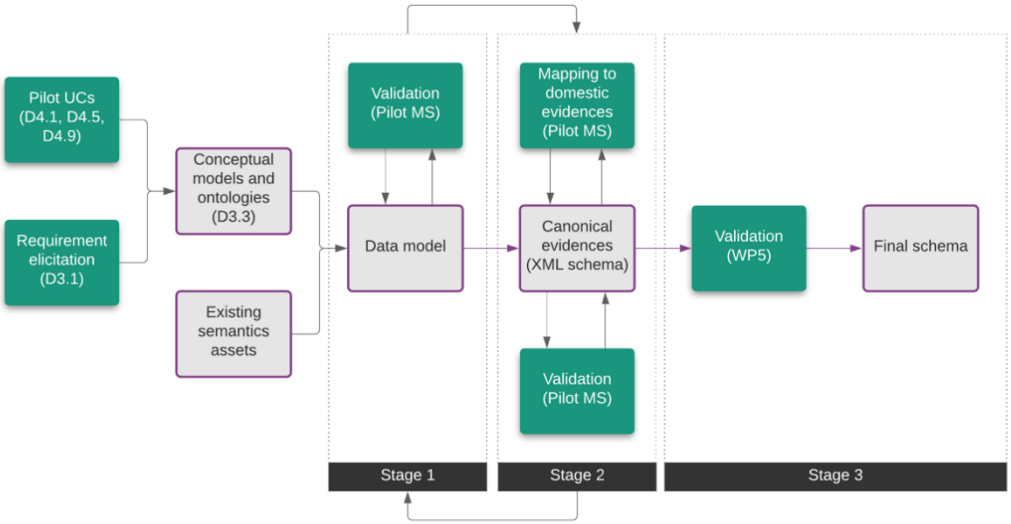


Figure 9: The canonical evidences development process

Accordingly, the preliminary data models are designed based on the existing resources in Stage 1 of the process. The primary models are tested with the piloting member state for model accuracy with respect to the attributes, cardinalities and the use of existing vocabularies and code lists. This was an iterative process followed by close dialogue with the respective pilot teams. The XML schemas for the models are developed in Stage 2 in close collaboration with the pilot partners and finally the schemas are validated during the integration into the technical system.

3.2 Pilot Studying abroad related Canonical Evidence

The Studying Abroad pilot has three use cases for which one common model is being developed as an application profile[14]. For each of the use cases subsets of the application will be used for constructing either XML schemas or JSON-LD based JSON schemas.

The three use cases are:

- Use Case 1: Application to public higher education
- Use Case 2: Applying for study grant
- Use Case 3: Diploma-Certs-Studies-Professional Recognition

[14] https://op.europa.eu/en/web/eu-vocabularies/application-profiles

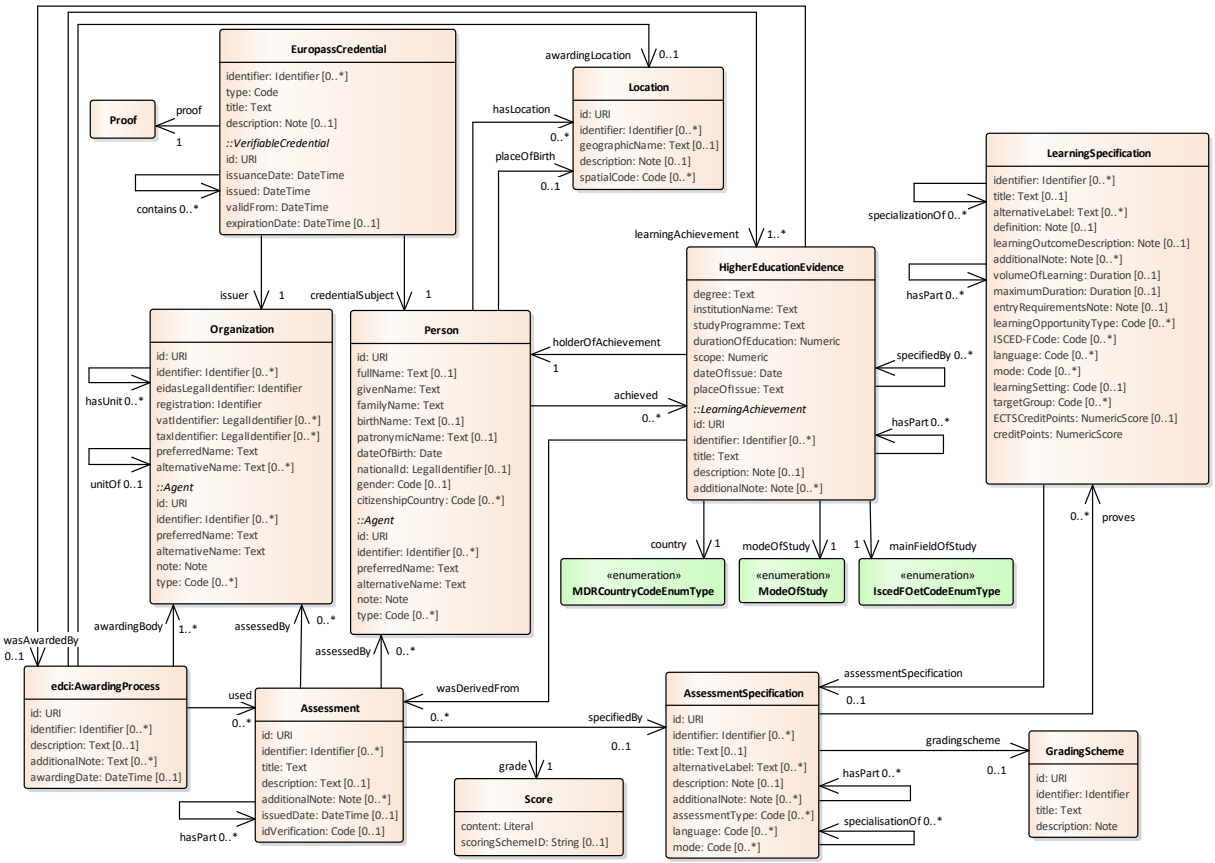


Figure 10: UML diagram of studying abroad application profile

At this point in time, it is only for use case 1 that a XML schema has been created.

The majority of elements in the application profile are classes and properties defined by the European Learning Model (EDCI) with just two classes and some properties defined and added for the purpose of Studying Abroad. The application profile for the Studying Abroad pilot, is more fully documented in Annex I.

3.2.1 Proof of completion of Higher Education

The model used for use case 1 contains both elements from EDCI and the classes and properties added by the Studying Abroad project.

The model reuses the class 'Person' and the enumeration classes 'MDRCountryCodeEnumType' and 'IscedFOetCodeEnumType' from EDCI. For the purpose of the project the class 'HigherEducationEvidence' has been defined as a subclass of the classe 'LearningAchievement' from EDCI and a small enumeration class, 'ModeOfStudy', was added too.

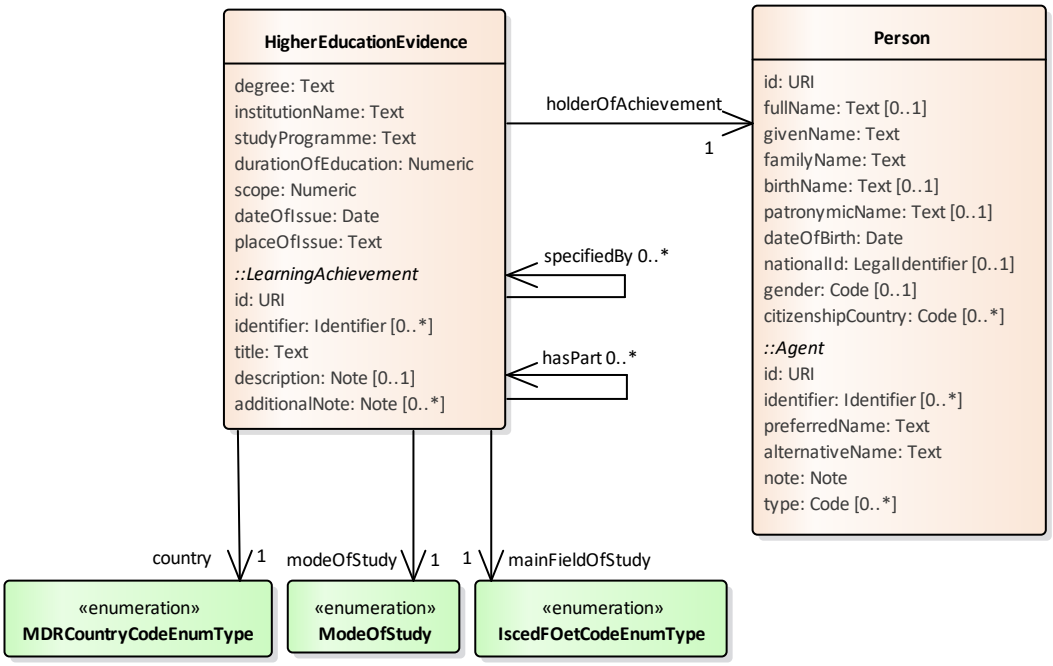


Figure 11: UML diagram for proof of completion of higher education

HigherEducationEvidence

Class description: The diploma data DCs need from a student.

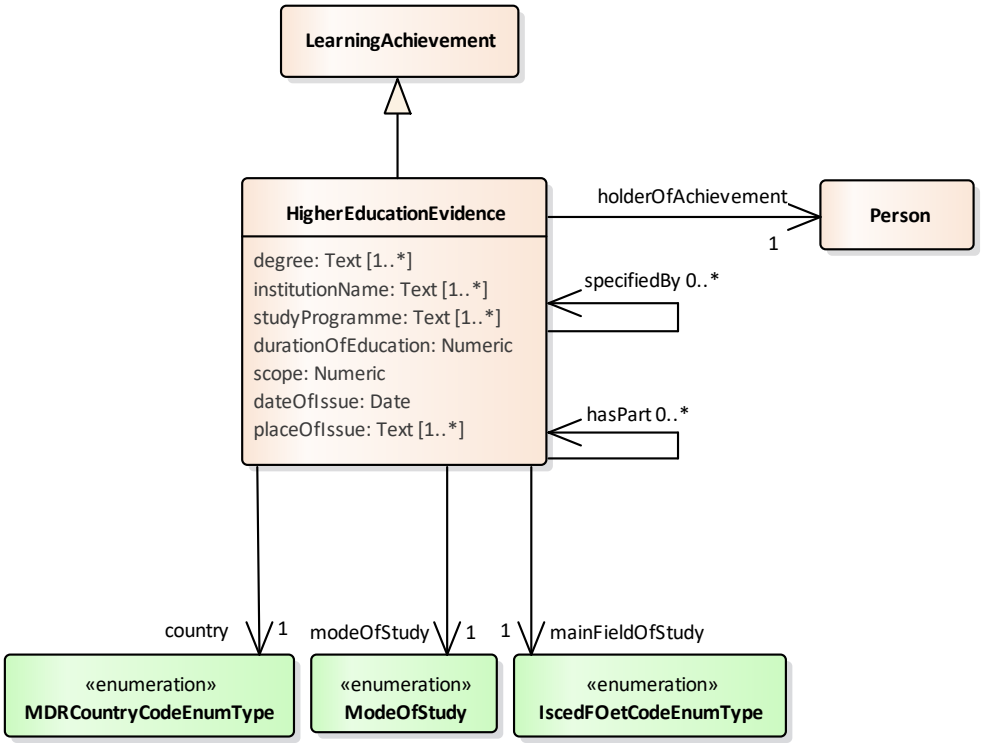


Figure 12: UML diagram for HigherEducationEvidence class

Table 3: HigherEducationEvidence class – definitions, data types and cardinalities

Label	Definition	Field	Range (data type)	Card
degree	An academic title or degree obtained by the student and proven by this diploma or certificate (evidence)	degree	Text	1 - *
institution name	The name of the higher education institution where the student obtained the degree	institutionName	Text	1 - *
study programme	Name of a study programme that the student finished at the higher education institution in order to obtain the degree	studyProgramme	Text	1 - *
duration of education	Official duration of education in years	durationOfEducation	Numeric	1
scope	The official workload of the study programme in the ECTS (European Credit Transfer and Accumulation System) credit points	scope	Numeric	1
date of issue	Date of issue of the certificate or diploma	dateOfIssue	Date	1
place of issue	Place of issue (location) of the certificate or diploma	placeOfIssue	Text	1 - *
country	Country where the study programme was completed by the student	country	MDRCountryCodeEnumType	1
main field of study	Field of finished higher education	mainFieldOfStudy	IscedFOetCodeEnumType	1
mode of study	Mode of study, (full time, part time, distance learning)	modeOfStudy	ModeOfStudy	1

ModeOfStudy (enumeration)

Class description: Enumeration class for the mode of study, Contains three enumerations: ‘full time’, ‘part time’ and ‘distance learning’.

Table 4: HigherEducationEvidence class – definitions, data types and cardinalities

Name	Type
distance learning	xsd:string

Name	Type
full time	xsd:string
part time	xsd:string

3.2.2 Proof of completion of Secondary education

A subset of the application profile will be selected, and a relevant schema will be created as part of the second iteration.

3.3 Pilot Doing Business Abroad Related Canonical Evidence

Within the scope of the piloting activities in DE4A project, the doing business abroad pilot (DBA) focusses on two use cases each of which are the most popular on cross boarder information exchange, namely, 1) Starting a business in another Member State (UC#1) and, 2) Doing business in another Member State (notify and update company data, unsubscribe – UC#2).

3.3.1 Company information

For the 1st iteration, the CompanyRegistration canonical evidence is the only evidence the DBA pilot requires to exchange information across borders in both the use cases. CompanyRegistration evidence is derived from the ontology presented in Figure 14, Section 5.4.2, in deliverable D3.3 "Semantic Framework – Initial Version". The DBA ontology presented therein included the attributes required for procedures covering UC1 and UC2. Furthermore, Annex XI of the same deliverable provides the details of the DE4A concepts and the related mappings to existing vocabularies for the identified attributes. Accordingly, in the initial version, the naming conventions for DE4A concepts were based on the TOOP RegisteredOrganisation ontology, the ISA^2 Core Vocabularies and pilot required data reported in D4.5.

For the canonical evidence, the same naming for the attributes is maintained. Attribute descriptions, cardinalities of the attributes, as well as the data types were modified and updated based on the requirements of the countries of the pilot representatives.

During the first iteration of the pilots (with the MVP) DBA is using basic data types, hardcoded data and a simple version of the canonical evidence. Therefore, complex data types and code lists were omitted in the current version of the DBA CompanyRegistration evidence.

Following the decisions made in the DE4A technical group, the canonical evidence is implemented in XSD format using XML Schema and the W3C namespace. The canonical evidence has the following main classes

- The legal entity (company) and its branches
- The location, address and the contact point of the company
- Company's legal name (either in English or in a national language of MS)
- The information of zero or more branches of the company

Table 5 below presents the classes and attributes, their data types and cardinalities of the CompanyRegistration evidence.

Table 5: Company information evidence – definitions, data types and cardinalities

DE4A attribute name	Definition	Type	cardinality
LegalEntity		Class	1
CompanyName	This is the primary name of the company. Can be provided for multiple languages	NamesType	[1..*]
CompanyType	Type of the company based on ISO 20275 (e.g: SA, PLC, LLC, GmbH etc)	string	[1..1]
CompanyStatus	Company status as defined in BRIS (closed, struck off the register, wound up, dissolved, economically active or inactive)	string	[1..1]
CompanyActivity	The activity of a company	ActivityType	[1..1]
RegistrationDate	Date of registration of the company	date	[1..1]
CompanyEndDate	The company end date	date	[0..1]
CompanyEUID	Identification of the company following the BRIS-structure: country code + register identifier + registration number + verification digit (optional)	string	[1..1]
VatNumber	The VAT registration number of the company	string	[0..*]
CompanyContactData	The contact information of the company (email and Telephone)	ContactPointType	[0..1]
RegisteredAddress	Links a Legal Entity to its registered address	AddressType	[1..*]
PostalAddress	Company physical address	AddressType	[0..*]
HasBranch	The branch information	BranchType	[0..1]
Names		Class	
LegalEntityName	Legal name of the company	string	[1..1]
ContactPoint		Class	
Email	A valid email address of the company	string	[0..*]
Telephone	Telephone number of the company	string	[0..*]
Activity		Class	
NaceCode	NACE-code of the company's activities	string	[0..*]
ActivityDescription	Description of the activity	string	[0..*]
Branch		Class	
BranchName	Primary name of the branch. Can be provided for multiple languages	NamesType	[1..1]

DE4A attribute name	Definition	Type	cardinality
BranchEUID	Identification of the branch of the company following the BRIS-structure: country code + register identifier + registration number + verification digit (optional)	string	[1..1]
BranchActivity	The activity of the branch presented by the NACE code and description. (NACE) codes are at EU level, however, the documentation claims that its compatible not only translantic but world level.	ActivityType	[0..1]
BranchRegistredAddress	The legal registered address of the branch	AddressType	[1..1]
BranchPostalAddress	The Physical address of the branch	AddressType	[0..1]
Address		Class	
PoBox	The Post Office Box number	string	[0..1]
Thoroughfare	The Street name	string	[0..1]
LocationDesignator	House number	string	[0..1]
PostCode	Postal code / zip code	string	[0..1]
PostName	City	string	[0..1]
AdminUnitL2	Administration unit Level2 - County / region / state	string	[0..1]
AdminUnitL1	Country	string	[0..1]

3.4 Pilot Moving Abroad Related Canonical Evidence

The main goal of the Moving Abroad pilot is to facilitate the evidence exchange when citizens are moving abroad. The benefits of this will be fewer physical movements of citizens to fetch the evidence and less interventions of civil servants resulting in a faster evidence exchange.

The benefits of DE4A will closely be related to WP5 outcomes piloted in WP4 "Common Component Design and Development", where multiple existing (e.g. eIDAS, e-Delivery) and new building blocks will be packaged for their pan-European adoption in the context of OOP and the SDG.

The Moving Abroad ontology presented in deliverable D3.3 "Semantic Framework – Initial version" included the attributes required for procedures covering the UC1 and UC2:

- UC1: Registering a change of address (basic registers thematic) – covered by Proof of Residency
- UC2: Requesting civil status certificates (population registration thematic) – covered by Birth and Marriage Certificates.

As mentioned in deliverable D3.3, Chapter 5, and in deliverable D4.9 "Moving Abroad - Use Case Definition & Requirements", the evidences of UC1 and UC2 are based on the structure public documents foreseen in the Regulation (EU) 2016/1191[15] of the European Parliament and of the Council of 6 July 2016 on promoting the free movement of citizens by simplifying the requirements for presenting certain public documents in the European Union and amending Regulation (EU) No 1024/2012. Towards representation of standardized forms for the Regulation on Public Documents, ISA developed XML Schema files (XSD). In order to maximize semantic and technical interoperability, these models use existing standards like the ISA[2] Core Vocabularies and the Universal Business Language (UBL), and include a subset of common mandatory attributes between EU Member States and other optional attributes that are MS-specific. UBL

[15] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R1191

is the first standard implementation of ebXML Core Components Technical Specification. The public documents that are related to DE4A pilot use cases are the Birth form (UC2), the Marriage form (UC2) and the Domicile and/or Residence form (UC1).

As a starting point, for the purpose of the MVP for the first iteration, the project will implement the canonical evidence types that facilitate the exchange of the minimum set of common attributes between EU Member States that are required for the procedure.

However, directly reusing the ISA XSD files for public documents for the canonical evidence is not feasible. The main reason is that these schemas are about representing PDF or XML forms but not evidence, violating the DE4A data minimisation principle for cross-border evidence exchange. These schemas contain metadata for the document (e,g, header, footer and document information) that is not directly needed for the procedure. Furthermore, apart from the common mandatory attributes identified by ISA, there are also optional attributes (e.g. parent details in birth certificate) with different naming conventions in each country but referring to the same concept. Therefore, it has been decided to reuse the current version of evidence data models by SDG Work Package 4 - Data Semantics, Formats & Quality[16] that defined common data formats for evidence types based on the EU Regulation for Public Documents. These formats for evidence types include the birth and marriage certificates that are related to UC2 for moving abroad. After an extensive study on the SDG models and ISA XSD files, it was observed that the attributes of SDG models are indeed a subset of the ISA XSD files for Public Documents.

For the canonical evidence, the same attribute naming conventions to the SDG models with the same data types (based on the ISA^2 Core Vocabularies) and cardinalities are used. During the first iteration of the pilots (with the MVP) the pilots are using basic data types, hardcoded data and a simple version of the canonical evidence. Therefore, complex data types and code lists were omitted in the current version of the evidences.

Following the recommendations for the DE4A technical implementation, the canonical evidence is implemented in XSD format using the XML Schema and the W3C namespace.

3.4.1 Birth Certificate

For the birth certificate, it is reused the current version (v0.14) of the respective SDG model. Apart from the representation of PDF and XML documents by ISA XSD files, the main difference between the SDG model for the birth certificate and the ISA XSD files is the common optional attributes identified by SDG. Moreover, there are some minor differences on the structure of the evidence (e.g. the birth event in SDG model includes the parent and the child as object properties while in the ISA XSD files the birth event includes only the birth person as an object property and the parent is an object property of the birth person) but the concepts remain the same.

Figure 13 depicts the current version (v0.14) of the diagram for Birth Evidence model. The models are analysed in Table 6 where classes and properties are presented with their data types and cardinalities.

[16] https://github.com/SEMICeu/SDG-sandbox

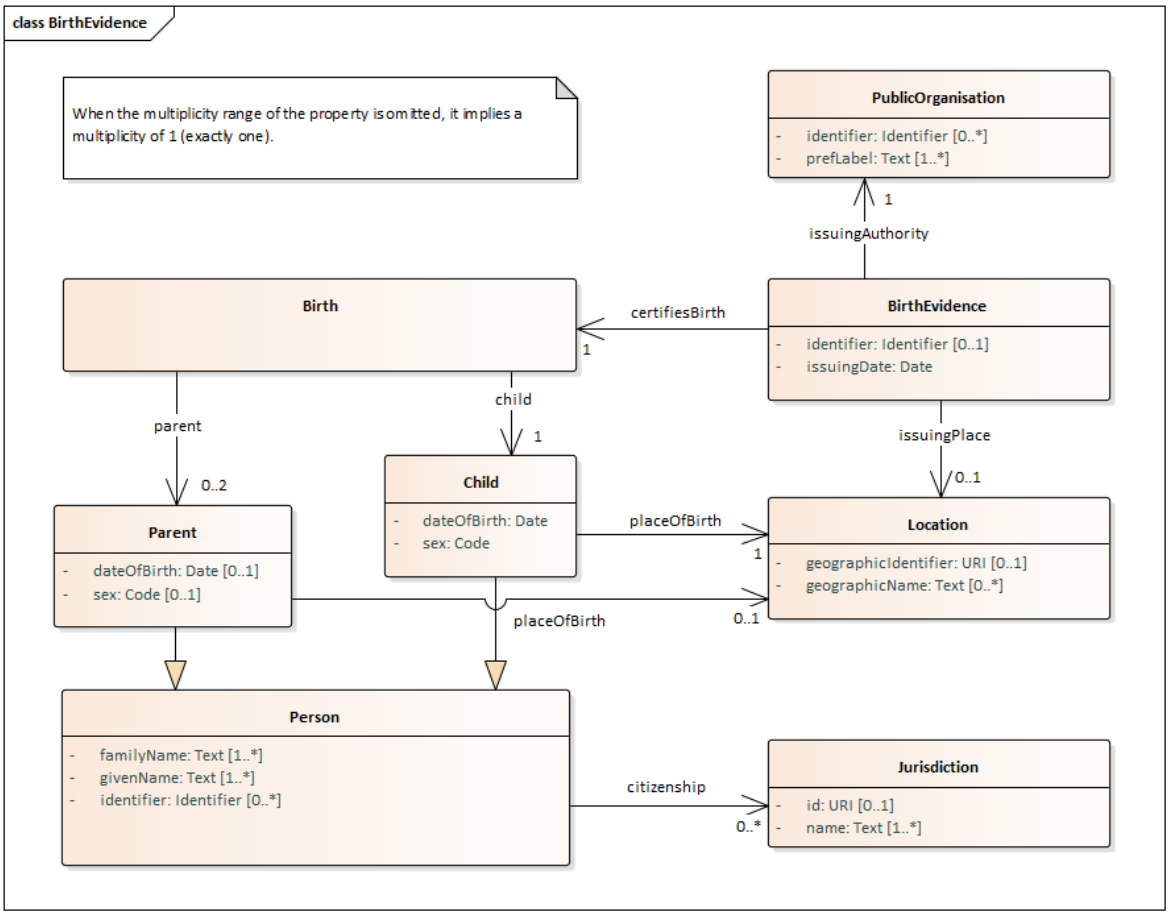


Figure 13: The diagram for Birth Evidence model (v0.14)[17]

Table 6: The table for Birth Evidence model

DE4A attribute name	Definition	Type	cardinality
BirthEvidence	This class contains elements related to the Birth Evidence.	BirthEvidence Type	
BirthEvidenceType			
Identifier	An unambiguous reference to the Birth Evidence.	udt:IdentifierType	[0..1]
IssueDate	The date on which the Birth Evidence was issued.	udt:DateType	[1..1]
IssuingAuthority	This class contains elements related to the issuing authority of the birth certificate	PublicOrganisationType	[1..1]
IssuingPlace	The Location where the Birth Evidence was issued.	LocationType	[0..1]
CertifiesBirth	Attesting in a formal way that the Birth is true.	BirthType	[1..1]
PublicOrganisationType	This class contains elements related to the issuing authority of the birth certificate		
Identifier	An organisation acronym or some other identifier.	cvb:LegalEntityIDType	[0..*]
PrefLabel	A preferred label is used to provide the primary, legally recognised name of the organisation. An organisation may only have one such name in any	cvb:LegalEntityLegalNameType	[1..*]

[17] https://github.com/SEMICeu/SDG-sandbox/blob/master/evidences/birth_certificate/data_model/

DE4A attribute name	Definition	Type	cardinality
	given language. Primary names may be provided in multiple languages with multiple instances of the preferred label property.		
LocationType	This class contains elements related to the location		
GeographicIdentifier	A URI that identifies the Location.	cvb:LocationGeographicIDType	[0..1]
GeographicName	A geographic name is a proper noun applied to a spatial object. The INSPIRE Data Specification on Geographical Names [INGN] provides a detailed model for describing a 'named place', including methods for providing multiple names in multiple scripts.	cvb:LocationGeographicNameType	[0..*]
BirthType	This class contains elements related to the birth event		
Child	The Person who is born at the Birth.	ChildType	[1..1]
Parent	The Parent of the Child.	ParentType	[0..2]
ChildType	This class contains elements related to the birth child		
DateOfBirth	The day on which the Child was born.	cvb:PersonBirthDateType	[1..1]
Sex	The chromosomal state, and reproductive organs and structures of a Person that allows them to be distinguished as female or male or undetermined.	cls:SexType	[1..1]
PlaceOfBirth	The Location where the Child was born.	LocationType	[1..1]
CountryOfBirth	The country where the Child was born.	CountryType	[1..1]
ParentType	This class contains elements related to the birth parent		
DateOfBirth	The day on which the Parent was born.	cvb:PersonBirthDateType	[0..1]
Sex	The chromosomal state, and reproductive organs and structures of a Person that allows them to be distinguished as female or male or undetermined.	cls:SexType	[0..1]
PlaceOfBirth	The Location where the Parent was born.	LocationType	[0..1]
CountryOfBirth	The country where the Parent was born.	CountryType	[0..1]
PersonType	This class contains elements related to the person		
FamilyName	A family name is usually shared by members of a family. This attribute also carries prefixes or suffixes which are part of the family name. Multiple family names are recorded in the single family name field.	cvb:PersonFamilyNameType	[1..*]
GivenName	A given name, or multiple given names, are the denominator(s) that identify an individual within a family. All given names are ordered in one field.	cvb:PersonGivenNameType	[1..*]
Identifier	The identifier relation is used to link a Person to any formally issued Identifier for that Person.	cvb:PersonIDType	[0..*]
Citizenship	The citizenship relationship links a Person to a Jurisdiction that has conferred citizenship rights on the individual such as the right to vote, to receive	JurisdictionType	[0..*]

DE4A attribute name	Definition	Type	cardinality
	certain protection from the community or the issuance of a passport.		
JurisdictionType	This class contains elements related to the jurisdiction		
ID	The value for the id property is a URI for that Jurisdiction.	cvb:JurisdictionIDType	[0..1]
Name	The name is simply a string that identifies the Jurisdiction, typically a country, with or without a language tag.	cvb:JurisdictionNameType	[1..*]
CountryType	This class contains elements related to a country.		
Code	ISO two-letter country code	ISOTwoletterCountryCodeIdentifierContentType	[0..1]

3.4.2 Marriage Certificate

For the marriage certificate, the current version (v0.14) of the respective SDG model is reused[18]. Apart from the representation of PDF and XML documents by ISA XSD files, the main difference between the SDG model for the marriage certificate and the ISA XSD files is the common optional attributes identified by SDG. Moreover, there are some minor differences on the cardinalities, where SDG identifies that in specific countries some attributes are considered optional (e.g. placeOfMarriage, familyNameAfterMarriage, familyNameBeforeMarriage) but the concepts remain the same.

Figure 14 depicts the current version (v0.14) of the diagram for Marriage Evidence model. The models are analysed in Table 7 where classes and properties are presented with their data types and cardinalities.

[18] https://github.com/SEMICeu/SDG-sandbox/tree/master/evidences/marriage_certificate/data_model

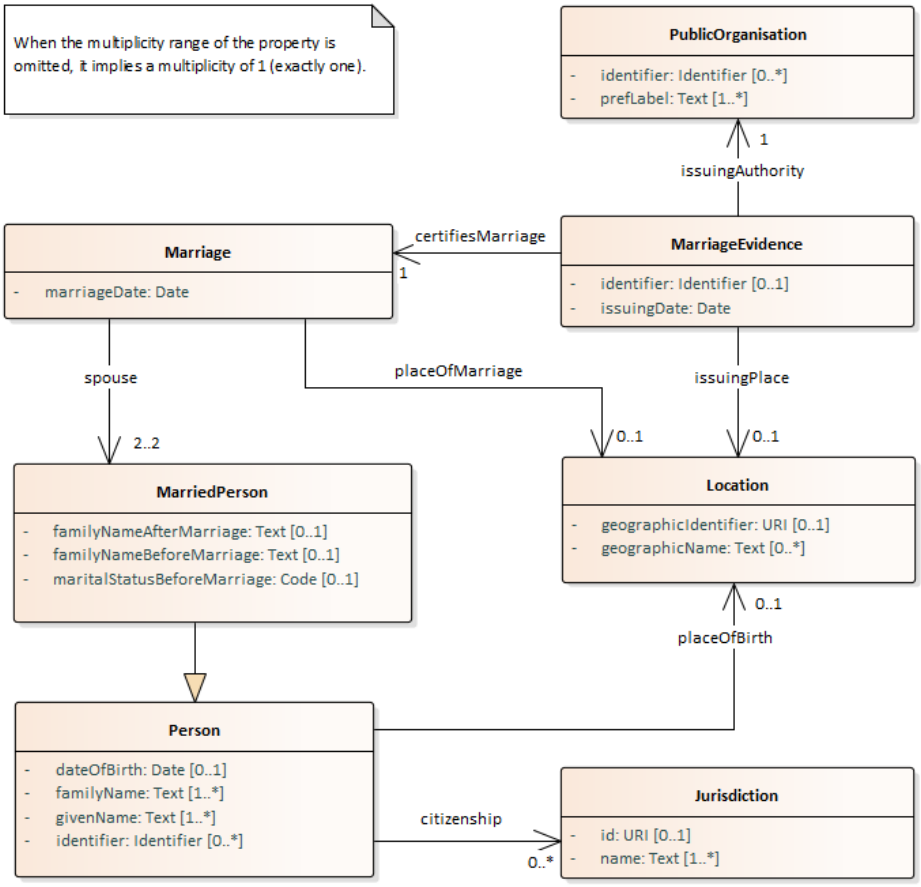


Figure 14: The diagram for Marriage Evidence model (v0.14)[19]

Table 7: The table for Marriage Evidence model

DE4A attribute name	Definition	Type	cardinality
MarriageEvidence	This class contains elements related to the Marriage evidence based on SDG	MarriageEvidenceType	
MarriageEvidenceType			
Identifier	An unambiguous reference to the Marriage Evidence.	udt:IdentifierType	[0..1]
IssueDate	The date on which the Marriage Evidence was issued.	udt:DateType	[1..1]
IssuingAuthority	A Public Organisation with official authority in charge of issuing the Marriage Evidence.	PublicOrganisationType	[1..1]
IssuingPlace	The Location where the Marriage Evidence was issued.	LocationType	[0..1]
CertifiesMarriage	Attesting in a formal way that the Marriage is true.	MarriageType	[1..1]
PublicOrganisationType	This class contains elements related to the issuing authority of the birth certificate	Class	
Identifier	An organisation acronym or some other identifier.	cvb:LegalEntityIDTyp	[0..*]

[19] https://github.com/SEMICeu/SDG-sandbox/tree/master/evidences/marriage_certificate/data_model

DE4A attribute name	Definition	Type	cardinality
PrefLabel	A preferred label is used to provide the primary, legally recognised name of the organisation. An organisation may only have one such name in any given language. Primary names may be provided in multiple languages with multiple instances of the preferred label property.	cvb:LegalEntityLegalNameType	[1..*]
LocationType	This class contains elements related to the location		
GeographicIdentifier	A URI that identifies the Location.	cvb:LocationGeographicIDType	[0..1]
GeographicName	A geographic name is a proper noun applied to a spatial object. The INSPIRE Data Specification on Geographical Names [INGN] provides a detailed model for describing a 'named place', including methods for providing multiple names in multiple scripts.	cvb:LocationGeographicNameType	[0..*]
MarriageType	This class contains elements related to marriage		
MarriageDate	The date on which the Marriage took place.	udt:DateType	[1..1]
Spouse	The Person who was married.	MarriedPersonType	[2..2]
PlaceOfMarriage	The Location where the Marriage took place.	LocationType	[0..1]
MarriedPersonType	This class contains elements related to the spouse		
FamilyNameAfterMarriage	This property contains the family name after the Marriage of the Person.	udt:TextType	[0..1]
FamilyNameBeforeMarriage	This property contains the family name before the Marriage of the Person.	udt:TextType	[0..1]
MaritalStatusBeforeMarriage	Situation with regard to whether a Person was single, married, separated, divorced or widowed.	udt:TextType	[0..1]
PersonType	This class contains elements related to the spouse		
DateOfBirth	The day on which the Person was born.	cvb:PersonBirthDateType	[0..1]
FamilyName	A family name is usually shared by members of a family. This attribute also carries prefixes or suffixes which are part of the family name. Multiple family names, are recorded in the single family name field.	cvb:PersonFamilyNameType"	[1..*]
GivenName	A given name, or multiple given names, are the denominator(s) that identify an individual within a family. All given names are ordered in one field.	cvb:PersonGivenNameType	[1..*]
Identifier	The identifier relation is used to link a Person to any formally issued Identifier for that Person.	cvb:PersonIDType	[0..*]
PlaceOfBirth	The Location where the Person was born.	LocationType	[0..1]
Citizenship	The citizenship relationship links a Person to a Jurisdiction that has conferred citizenship rights on the individual such as the right to vote, to receive certain protection from the community or the issuance of a passport.	JurisdictionType	[0..*]
JurisdictionType	This class contains elements related to the jurisdiction		

DE4A attribute name	Definition	Type	cardinality
ID	The value for the id property is a URI for that Jurisdiction.	cvb:JurisdictionIDType	[0..1]
Name	The name is simply a string that identifies the Jurisdiction, typically a country, with or without a language tag.	cvb:JurisdictionNameType	[1..*]

3.4.3 Residence Proof

For the proof of residence, there is no SDG model available. The relevant information is provided by the Domicile and/or Residence form from Annex X of EU Regulation for Public Documents. Nevertheless, it is observed that the common attributes (mandatory and optional with different naming conventions) share many similarities with the Birth and Marriage certificates. Therefore, it was decided to use the same structure with Birth and Marriage certificates so that all canonical evidence types in the Moving Abroad pilot share the same structure and to align as much as possible with the SDG and ISA XSD files.

The models are analysed in Table 8, based on the identified concepts by ISA XSD files and the structute of SDG evidence data models, where classes and properties are presented with their data types and cardinalities. Note the many optional elements to compensate for the fact that not all data is obtainable in all countries.

Table 8: The table for Residence Proof model

DE4A attribute name	Definition	Type	cardinality
ResidenceProof	This class contains elements related to the Residence Proof.	ResidenceProofType	[0..1]
ResidenceProofType			
Identifier	An unambiguous reference to the Residence Proof.	udt:IdentifierType	[0..1]
IssueDate	The date on which the Birth Evidence was issued.	udt:DateType	[1..1]
IssuingAuthority	This class contains elements related to the issuing authority of the Residence Proof.	PublicOrganisationType	[1..1]
IssuingPlace	The Location where the Residence Proof was issued.	LocationType	[0..1]
CertifiesResidence	Attesting in a formal way that the Residence is true.	ResidenceType	[1..1]
PublicOrganisationType	This class contains elements related to the issuing authority of the birth certificate		
Identifier	An organisation acronym or some other identifier.	cvb:LegalEntityIDType	[0..*]
PrefLabel	A preferred label is used to provide the primary, legally recognised name of the organisation. An organisation may only have one such name in any given language. Primary names may be provided in multiple languages with multiple instances of the preferred label property.	cvb:LegalEntityLegalNameType	[1..*]
LocationType	This class contains elements related to the location		
geographicIdentifier	A URI that identifies the Location.	cvb:LocationGeographicIDType	[0..1]

DE4A attribute name	Definition	Type	cardinality
geographicName	A geographic name is a proper noun applied to a spatial object. The INSPIRE Data Specification on Geographical Names [INGN] provides a detailed model for describing a 'named place', including methods for providing multiple names in multiple scripts.	cvb:LocationGeographicNameType	[0..*]
ResidenceType	This class contains elements related to the domicile-residence and/or previous domicile-residence		
Inhabitant	Person, living in the residence or domicile	PersonType	[1..1]
Residence	Current residence inhabited by person	AddressType	[0..1]
Domicile	Current domicile inhabited by person	AddressType	[0..1]
PreviousResidence	Previous residence inhabited by person	AddressType	[0..1]
PreviousDomicile	Previous domicile inhabited by person	AddressType	[0..1]
PersonType	This class contains elements related to the person		
FamilyName	A family name is usually shared by members of a family. This attribute also carries prefixes or suffixes which are part of the family name. Multiple family names are recorded in the single family name field.	cvb:PersonFamilyNameType	[1..*]
GivenName	A given name, or multiple given names, are the denominator(s) that identify an individual within a family. All given names are ordered in one field.	cvb:PersonGivenNameType	[1..*]
Identifier	The identifier relation is used to link a Person to any formally issued Identifier for that Person.	cvb:PersonIDType	[0..*]
DateOfBirth	The day on which the person was born.	cvb:PersonBirthDateType	[0..1]
Sex	The chromosomal state, and reproductive organs and structures of a Person that allows them to be distinguished as female or male.	cls:SexType	[0..1]
MaritalStatus	An indicator of the marital status	udt:TextType	[0..1]
PlaceOfBirth	The Location where the person was born	LocationType	[0..1]
CountryOfBirth	The Country where the person was born.	CountryType	[0..1]
Citizenship	The citizenship relationship links a Person to a Jurisdiction that has conferred citizenship rights on the individual such as the right to vote, to receive certain protection from the community or the issuance of a passport.	JurisdictionType	[0..*]
JurisdictionType	This class contains elements related to the jurisdiction		
ID	The value for the id property is a URI for that Jurisdiction.	cvb:JurisdictionIDType	[0..1]
Name	The name is simply a string that identifies the Jurisdiction, typically a country, with or without a language tag	cvb:JurisdictionNameType	[1..*]
AddressType	This class contains elements related to an address		
FullAddressOrPOBox	This property contains contains the full address or Street PO Box details of the address.	cvb:AddressFullAddressType	[1..1]
PostCode	This property contains the location specification of the address.	cvb:AddressPostCodeType	[0..1]

DE4A attribute name	Definition	Type	cardinality
Country	The Country where the person was born.	CountryType	[0..1]
CountryType			
CountryType	This class contains elements related to a country.		
Code	ISO two-letter country code	ISOTwoletterCountryCodeIdentifierContentType	[0..1]

4 Information Desk Specification

The Information Desk (IDK), which was initially introduced in D3.3, constitutes a key semantic asset within DE4A that facilitates cross-border evidence exchange between Data Consumers (DCs) and Data Providers (DPs). Through the use of the IDK, DCs and DPs can obtain the required information in order to submit requests and/or responses to the respective stakeholders. This chapter presents the first iteration of the implemented IDK specification, describing the process for designing and developing the underlying semantic model, and also provides an overview of the initial specification of the Application Programming Interface (API) for accessing information residing in the model. The chapter concludes with directions towards implementing the final version of the model and API.

4.1 IDK Semantic Model

The first iteration of the IDK semantic model takes the form of an OWL 2 DL[20] ontology and its design is based on the respective requirements presented in D3.3, Chapter 5. The process for developing the IDK ontology is in accordance with the established ontology engineering methodology presented in [16], as described in the following subsections.

4.1.1 Determining the Domain and Scope

Within ontology engineering, a typical method for outlining the scope of knowledge represented by an ontology is through a set of Competency Questions (CQs), namely, natural language queries that express a pattern for a type of question the ontology should be able to answer [17]. In essence, CQs represent functional requirements, in the sense that the developed ontology or the respective ontology-based system(s) should be able to respond to. If this is the case, then the ontology and respective system(s) are considered as containing all the relevant knowledge. Thus, CQs are used both for the specification, as well as for the validation of the ontology. In this context, Table 9 shows the CQs that drive the design of the IDK semantic model; the CQs were extracted from the respective requirements presented in D3.3, Chapter 5, as well as on the architecture, pilots and common technical components needs.

Table 9: Competency Questions for the IDK semantic model.

CQ#	Competency Question	Sample Responses
CQ1	What are the types of canonical evidence?	HigherEdCertificate, SecondaryEdCertificate, BirthCertificate, etc.
CQ2	How is a canonical evidence type identified?	With a URN, according to the DE4A policy for identifiers, like, e.g., urn:eu-de4a:CanonicalEvidenceType::HigherEdCertificate
CQ3	What are the different levels of administrative territorial units?	National (NUTS0), regional (NUTS1-3), local (LAU) and educational (EDU).
CQ4	How is an evidence provision identified?	By the canonical evidence type provided and the data owner who provides it.
CQ5	How many types of canonical evidence does an evidence provision involve?	Only one.
CQ6	What canonical evidence type does evidence provision [X] provide?	E.g., CompanyRegistration.
CQ7	How are Data Owners (DOs) identified?	With a URN, according to the DE4A policy for identifiers, like, e.g., iso6523-actorid-upis::9991:PT990000101

[20] https://www.w3.org/TR/owl2-profiles/

CQ#	Competency Question	Sample Responses
CQ8	How many DOs are responsible for an evidence type [X]?	Only one at a specific administrative territorial unit (ATU). Thus, if there are two DOs responsible of [X] at ATU[Y] and ATU[Z], and the latter is in a lower administrative level than the former (e.g., a region within a country), only the DO at highest administrative level (ATU[Y]) can be included as responsible of [X].
CQ9	Who is the DO that provides evidence type [X]?	E.g., iso6523-actorid-upis::9991:PT990000101
CQ10	What are the administrative levels in country [X]?	National (NUTS0), the regional levels (NUTS1-3) that correspond to public administrations and not only statistical regions, local (LAU) and educational (EDU).
CQ11	How many administrative territorial units can have administrative level [X]?	Only one if the administrative level is NUTS0; otherwise, one or more.
CQ12	How many administrative levels can correspond to administrative territorial unit [X]?	Only one.
CQ13	How many administrative levels can have administrative territorial units that provide evidence type [X] in country [Y]?	Only one.
CQ14	How many administrative territorial units are associated to a DO?	Only one that corresponds to the DO competence territorial scope.
CQ15	What is the administrative level of the DOs that issue evidence type [X]?	E.g., NUTS2.
CQ16	How many countries are involved in the provision of evidence service [X]?	Only one.
CQ17	How many types of evidence provisions are there?	Two types: USIP provision & IP Provision. IP provision has two sub-types: Proxy Provision and Direct Provision.
CQ18	What are the differences between the types of evidence provisions?	USIP provision MUST include a URL to redirect the user to the DO Portal; IP provision MAY include additional information as input parameters required by the service.
CQ19	What is the administrative territorial unit that delimits the scope of an evidence provision?	The ATU of the DO when the evidence provision is of type USIP provision or IP Direct Provision. When the type is Proxy Provision, more than one ATU can be involved and the final ATU source of the evidence will be set with an input parameter.
CQ20	What is the identifier of the evidence provision?	A URN composed by the corresponding canonical evidence type and data owner, like, e.g., urn:eu-de4a:provision::9991:PT000000029:HigherEdCertificate

4.1.2 Reusing Existing Resources

A typical step in ontology engineering involves considering the reuse of existing third-party resources while designing a semantic model. These resources can either be existing established ontologies "lending" a subset of their definitions to the ontology under development, or other types of resources, like, e.g., technical documentation, publicly available information, etc.

Therefore, for documenting the ontology concepts, the project relies on SKOS (Simple Knowledge Organization System), which is a popular common data model for sharing and linking knowledge

organization systems [18]. More specifically, concepts from the IDK ontology with the following SKOS documentation properties are annotated:

- `skos:prefLabel` – The preferred lexical label for a resource.
- `skos:altLabel` – An alternative lexical label for a resource. Acronyms, abbreviations, or spelling variants are typically annotated as alternative labels.
- `skos:definition` – A statement or formal explanation of the meaning of a concept.
- `skos:example` – An example of the use of a concept.
- `skos:editorialNote` – A note for an editor or maintainer of the vocabulary with pointers that may be included in the next release of the model.

Besides SKOS, a core inspiration for the IDK model is the ISA[2] BRegDCAT-AP specification[21] that provides a standard data model for specifying base registries access and interconnection. Also heavily relying on the ISA[2] Core Vocabularies, BRegDCAT-AP models an evidence as a dataset and an evidence service as a dataset distribution and serves as a basis for our definition of the "Dataset" concepts, as described in the following subsections.

4.1.3 Specifying the Key Concepts

The aim of this step is to come up with a list of the key concepts that will be included in the IDK model. At this stage, the project does not need to worry yet whether there is an overlap between concepts, what the relations among the concepts may be, or what properties the concepts may have, as these issues will be clarified during the next two steps. Furthermore, concepts do not necessarily refer to entities only, but they can also represent attributes or properties of entities.

Thus, based on D3.3 and on frequent consultations with DE4A project partners, the following core concepts for the IDK have been identified, along with a respective set of "business rules" – core concepts are annotated in **bold dark blue font** below:

Canonical Evidence Type: A canonical evidence type is an agreed dataset with a common data model that is an application profile of the corresponding domain ontology or vocabulary:

- According to the DE4A policy for identifiers, a canonical evidence type is **identified with a URN**, which includes the token name of such canonical evidence.
- A canonical evidence type is composed by a set of **data elements** according to its common data model.
- The information provided by a canonical evidence type dataset offers the same proof regardless the data owner that issues the data; otherwise, the dataset should be reorganised in different canonical evidence types (e.g., Diploma vs Course Results).
- The data provided according to the common data model defined for a canonical evidence type is called **canonical evidence**. The issuing authority must guarantee the equivalence of the proof provided by a canonical evidence and by the associated domestic evidence –evidence originally issued by competent authorities with legal value.

Evidence Provision: Availability of the provision of certain canonical evidence type by certain data owner:

- An evidence provision involves one and only one **canonical evidence type**.
- An evidence provision includes one and only one **Data Owner** (DO).
- An evidence provision is **identified with a URN** combining the identifier of the corresponding canonical evidence type and data owner, according to the DE4A policy for identifiers.
- If a canonical evidence type is provided by several data owners within a country, all of them belong to the same **administrative territorial level** (ATU).

[21] https://joinup.ec.europa.eu/collection/access-base-registries/solution/abr-bregdcat-ap

- The evidence provision uses either the Intermediation Pattern or the User Supported Intermediation Pattern (**USIP Provision**); in the first case, the evidence provision can be either a **Direct Provision** or a **Proxy Provision**.
- A USIP Provision must include the URL to redirect the user to the DO portal (**redirect URL**).
- An IP provision may include additional information required by the data owner to properly locate the requested evidence, i.e., for the record matching (**input parameters**).
- An evidence provision could involve more than an **administrative territorial level** (ATU) if it is a **Proxy Provision**.
- An IP Provision can specify alternative input parameter sets to properly locate the evidence (i.e., record matching), each of them identified by a **sequential number** and, alternatively, by a **short descriptive title**.
- An input parameter set is composed of a set of **data elements**.
- An evidence provision is implemented by at least one service endpoint created by a data transferor within the same country of the corresponding data owner. Service endpoints and data transferors are managed by SMP routing components and they are out of the scope of this document.

Data Element:

- A data element is identified by a **URI path** that represents the **hierarchical relationship** of the corresponding term within the ontology or general-purpose vocabulary.
- A data element can be an **entry of a list**, a **complex concept**, a **simple concept**, or a **property** of a complex or a simple concept that can be either a complex or a simple concept. These concepts determine the type of the data element.
- A data element is described by a **label** and a **description;** optionally an **example** of the term can be provided. Labels, descriptions and examples must be expressed in English and they can also be expressed in one or more other **languages**.
- The translation of the data element label/description to a specific language can be tagged as **verified** or **not verified**. The verification of such translation means that it can be considered an official translation.

Data Owner: public organisation, identified by an ISO6523 identifier, that is responsible of the provision of canonical evidences:

- The Data owner of a USIP or a Direct Provision is also the evidence issuing authority.
- A Proxy provision has a Data owner (DO) that is an organisation running an intermediation platform that hides the organisational complexity behind it. This platform provides proxy services that are either redirect services or choreography services that connect with services provided by the issuing authorities, so the final recipients of the evidence exchange network can only be proxy services.
- If needed, to properly locate the issuing authority (e.g., a regional public administration), a Proxy Provision can require the ATU at a given administrative level as an input parameter.

4.1.4 Defining the Classes and Class Hierarchy

Based on the set of concepts identified in the previous step, during this step the set of classes that reside in the first iteration of the IDK semantic model are defined (see table 10).

Table 10: List of classes and respective definitions in the IDK model.

Class Name	Definition
AdministrativeLevel	Administrative level of an administrative territorial unit. E.g., 'nuts0' for national level; 'nuts1', 'nuts2' or 'nuts3' for regional or province level; 'lau' for local level; and 'edu' for educational level. There are Member States where some regional levels are not public administrations (e.g., Spanish nuts1 is defined only by statistical purposes and has not any administrative meaning).

Class Name	Definition
AdministrativeTerritorialUnit	Administrative territorial unit as defined by the organization of the public administrations in each Member State according to their territorial scope of administrative competences.
CanonicalEvidenceType	Canonical evidence type that has associated a token name (e.g., BirthCertificate) and a common data model. The proof provided by the information relevant to a canonical evidence according to a canonical evidence type is equivalent regardless the issuing authority.
Country	Administrative territorial unit at national level, identified by an ISO 3166 Alpha 2 code.
EvidenceProvision	Evidence issuing availability by a specific Data Owner in correspondence to a specific canonical evidence type. An Evidence Provision can use either the intermediation pattern or the user-supported intermediation pattern; in the latter case, the URL to redirect the user to the DO portal must be specified, and in the former case, additional information can be required by the corresponding service as input parameters to properly locate the evidence (record matching). If the data owner of an evidence provision under the intermediation pattern is a proxy backend, then it is a Proxy Provision; otherwise, it is a Direct Provision.
InputParameterSet	Optional set of data elements that may be alternatively required by an evidence provision to properly locate the evidence, i.e., record matching, apart from the eIDAS identification of the evidence data subject.
MultilingualDescription	Description of data elements (label, description, example, verification) in a particular language.
OntologyDataElement	Data element that is part of a canonical evidence data model or an input parameter set. Can be of either a simple or a complex type.
PublicOrganization	Organization of a Public Administration with administrative competences.

4.1.5 Defining the Properties of Classes

The class properties offer additional information regarding the internal structure of entities, as well as their interrelationships with other entities in the model. This step is devoted to specifying the properties, where every property is associated to one or more classes (i.e., the property *domain*) and to a value type (i.e., the property *range*). Properties that assume raw data values, like, e.g., strings, integers, etc., are called *datatype properties*, while properties that express relationships between class objects (i.e., assume instances of other classes are their values) are called *object properties*.

Table 11 includes the list of properties and their definitions in the IDK model, along with their type (object vs data), domain and range.

Table 11: List of properties along with respective definitions, domain, and range in the IDK model.

Property Name	Definition	Type	Domain	Range
hasATU	The administrative territorial units comprising the country.	ObjectProperty	Country	AdministrativeTerritorialUnit
hasCode	The code of the administrative territorial unit.	Datatype Property	AdministrativeTerritorialUnit	string
hasCountryCode	ISO 3166 A2 country code.	Datatype Property	Country	string

Property Name	Definition	Type	Domain	Range
hasDataElement	The data elements comprising the input parameter set or the canonical evidence type.	ObjectProperty	CanonicalEvidenceType OR InputParameterSet	OntologyDataElement
hasDataOwner	The public organization that lawfully issues the evidence data that is transferred through the evidence provision, according to the ISO 6523 specification.	ObjectProperty	EvidenceProvision	PublicOrganization
hasDefinition	The definition of the multilingual description.	Datatype Property	MultilingualDescription	string
hasDescription	Multilingual descriptions of data elements.	ObjectProperty	OntologyDataElement	MultilingualDescription
hasInputParameterSet	Set of data elements that may be required by an evidence provision.	ObjectProperty	IPProvision	InputParameterSet
hasLabel	The label of the description.	Datatype Property	MultilingualDescription	string
hasLanguage	The language of the description.	Datatype Property	MultilingualDescription	string
hasLevel	The administrative level of an administrative territorial unit.	ObjectProperty	AdministrativeTerritorialUnit	AdministrativeLevel
hasParentATU	Hierarchical organization of administrative territorial units. Transitive property.	ObjectProperty	AdministrativeTerritorialUnit	AdministrativeTerritorialUnit
hasParentLevel	Hierarchical organization of administrative levels. Transitive property.	ObjectProperty	AdministrativeLevel	AdministrativeLevel
hasPath	The path of the administrative territorial unit.	Datatype Property	AdministrativeTerritorialUnit	string
hasRedirectURL	URL to redirect the user in the user-supported intermediation pattern.	Datatype Property	USIPProvision	URL
hasType	The type of the ontology data element.	DataProperty	OntologyDataElement	string
hasURN	Unique identifier of an entity. Canonical evidence and evidence provision are identified based on DE4A identifier policy, while public organizations are identified according to the ISO 6523 specification.	DataProperty	EvidenceProvision OR CanonicalEvidenceType OR PublicOrganization OR OntologyDataElement	string

Property Name	Definition	Type	Domain	Range
isInATU	The administrative territorial unit a public organization belongs to.	ObjectProperty	PublicOrganization	AdministrativeTerritorialUnit
isOptional	Indicates whether the specific data element should be present or not.	Datatype Property	OntologyDataElement	boolean
isOrganizedIn	The administrative levels a country is organized in.	ObjectProperty	Country	AdministrativeLevel
isProvidedBy	The evidence provision providing the canonical evidence type.	ObjectProperty	CanonicalEvidenceType	EvidenceProvision
isVerified	Shows whether a competent authority has verified that the label and description are correctly translated via an automatic translation.	Datatype Property	MultilingualDescription	boolean

4.2 IDK Ontology Overview

Figure 15 illustrates the key IDK classes and their interrelationships, using the Graffoo notation [19] – the yellow rectangles indicate the classes, while the blue arrows indicate the object properties (see previous subsection) linking classes together. In order to prevent the diagram from being overcomplicated, the illustration of data properties were omitted.

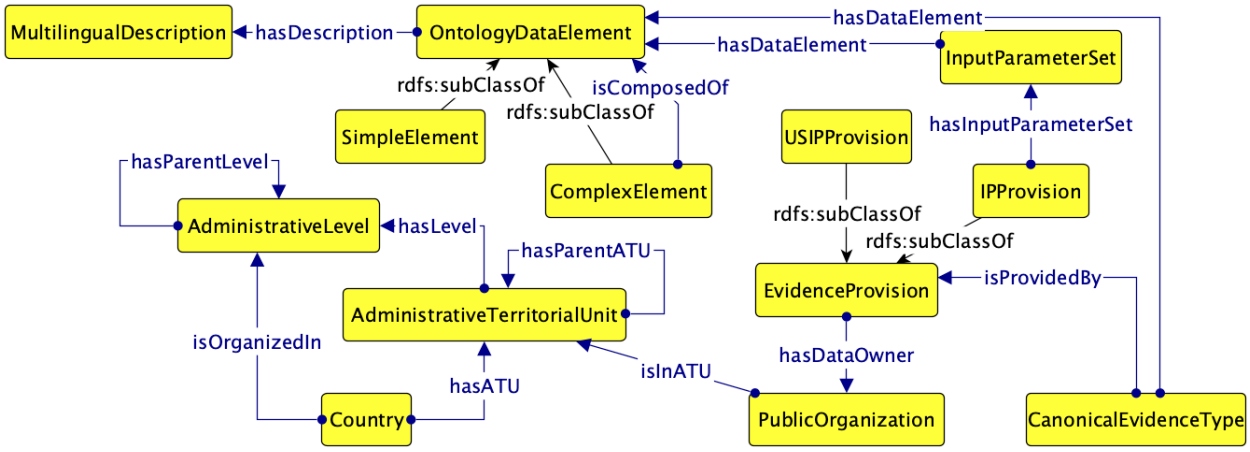


Figure 15: Overview of the IDK ontology classes and their interrelationships.

The following subsections illustrate sample instantiations from the model.

4.2.1 Representation of Administrative Territorial Units

Figure 16 demonstrates the representation of a set of Spanish ATUs along with their respective classifications, based on the NUTS 2021 classification[22]; *NUTS* (*Nomenclature of Territorial Units for Statistics*) is a hierarchical system for dividing up the EU economic territory.

[22] https://ec.europa.eu/eurostat/web/nuts/background

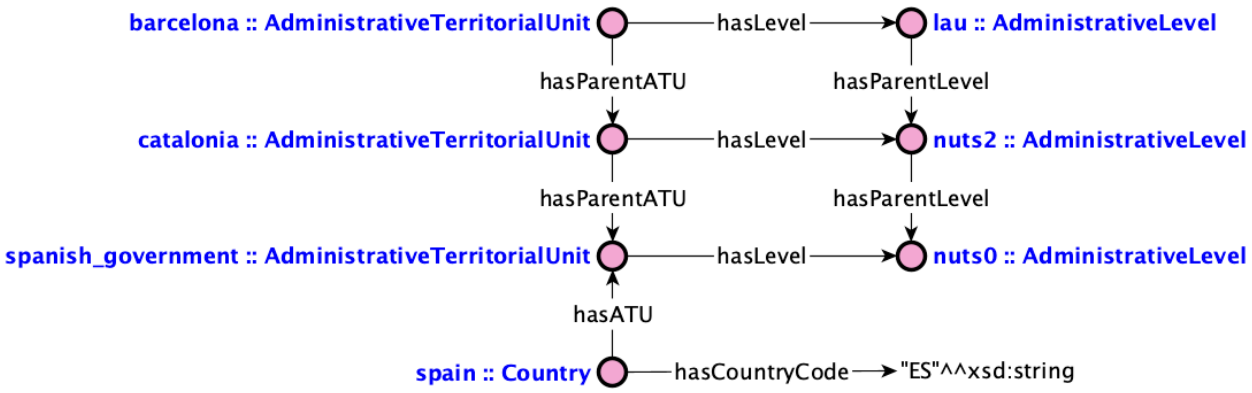


Figure 16: Sample set of Spanish ATUs and their classifications.

4.2.2 Representation of Evidence Provisions

Figure 17 illustrates a sample evidence provision with a data owner at national level that provides the company data canonical evidence type.

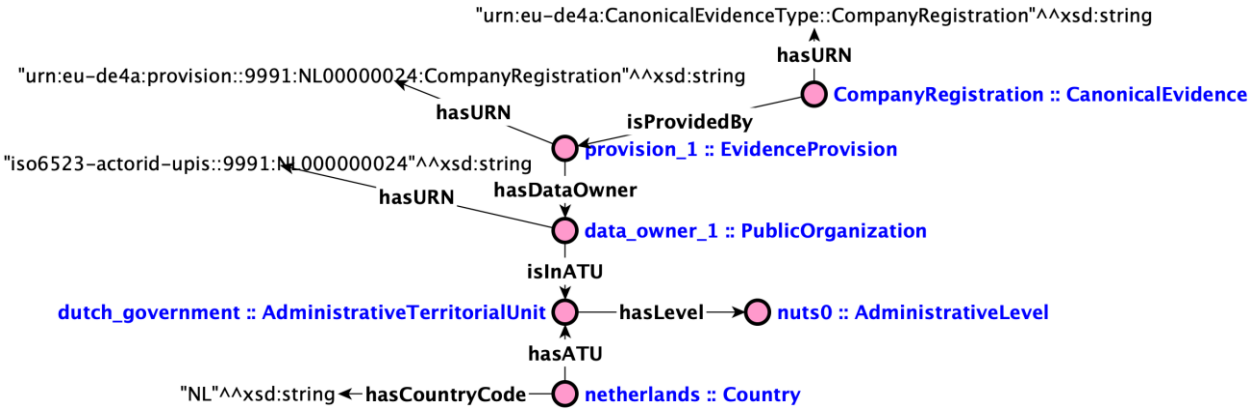


Figure 17: Sample evidence provisioninstantiation.

4.2.3 Representation of Data Elements and Multilingual Descriptions

In order for a canonical evidence to be understandable by all involved stakeholders, the IDK model encompasses the concepts of `OntologyDataElement`s and accompanying `MultilingualDescription`s that allow expressing corresponding labels of canonical evidence fields along with the respective definitions in a number of languages.

Thus, suppose that the following XSD fragment describes a piece of birth evidence:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
 <xs:element name="BirthEvidence">
  <xs:complexType>
   <xs:all>
    <xs:element name="CertifiedBirth">
     <xs:complexType>
      <xs:all>
       ...
       <xs:element name="DateOfBirth" type="xs:date"/>
       <xs:element name="PlaceOfBirth" type="xs:string"/>
       ...
      </xs:all>
     </xs:complexType>
    </xs:element>
   </xs:all>
  </xs:complexType>
 </xs:element>
 ...
</xs:schema>
```

Figure 18 illustrates the instantiation in the IDK ontology of the above data elements representing the place and date of birth, along with the respective multilingual descriptions in English and in Spanish.

Figure 18: Sample instantiation of a subset of data elements and respective multilingual descriptions comprising a birth certificate.

4.3 IDK Ontology Evaluation

This section presents the evaluation of the IDK ontology with respect to its consistency, quality, structure and compliance with the requirements.

4.3.1 Consistency and Quality Evaluation

First of all, an RDF/XML serialization of the ontology was submitted to the W3C RDF validator[23] and it validated successfully. Furthermore, in order to evaluate the IDK ontology's consistency and quality, *OOPS* (*OntOlogy Pitfall Scanner*) was used. This is a popular online tool for detecting the most common pitfalls in semantic models [20]. OOPS analyses a submitted ontology and provides a list with the detected pitfalls and respective negative consequences, followed by suggestions towards fixing the issues and improving the overall quality of the ontology. In a nutshell, OOPS can detect:

- *Critical pitfalls* affecting the ontology's consistency, which definitely need to be fixed;
- *Important pitfalls*, which are not equally critical, but ideally should be fixed;
- *Minor pitfalls*, which do not lead to any critical problems, but fixing them will improve the quality of the model.

We submitted the first iteration of the IDK ontology to OOPS, which detected the issues in Table 12.

Table 12: Issues and pitfalls detected by OOPS, along with our measures for fixing them.

Issue detected by OOPS	Measures to fix the issue
Minor – Missing annotations (e.g., rdfs:label, skos:prefLabel, skos:altLabel, rdfs:comment) for some classes and properties – 32 cases.	We added the missing annotations.
Important – Missing domain or range in properties – 14 cases.	This issue concerns only the imported SKOS ontology (see Subsection 4.1.2), thus, it was not addressed.
Minor – Inverse relationships not explicitly declared – 17 cases.	The suggested inverse relationships were not deemed necessary for the purposes of the first iteration of the IDK model but will be considered for the second and final version.
Important – No license declared.	The first iteration of the ontology does not yet have a release license. The final version will be released under an open license.

4.3.2 Evaluating the Structure of the Ontology

Structural parameters of the IDK ontology were evaluated by submitting it to OntoMetrics[24], an online tool that validates ontologies based on established metrics. Table 13 presents the generated results: (a) *Base Metrics* comprise simple metrics, like the count of classes, axioms, objects etc.; these metrics show the quantity of ontology elements; (b) *Schema* address the design of the ontology and indicate the richness, width, and depth of a semantic model.

[23] W3C RDF validator: https://www.w3.org/RDF/Validator/

[24] https://ontometrics.informatik.uni-rostock.de

Table 13: Ontology metrics for the first iteration of the IDK ontology, generated by OntoMetrics.

Base Metrics	Class count	11
	Object property count	12
	Datatype property count	9
	SubClassOf axioms count	2
	Disjoint classes axioms count	2
	Functional property axioms count	4
	Transitive object property axioms count	2
	DL expressivity	*ALU+(D)*
Schema Metrics	Attribute richness	0.818182
	Inheritance richness	0.181818
	Relationship richness	0.857143
	Axiom/class ratio	23.090909
	Class/relation ratio	0.785714

The base metrics Indicate that the IDK model is very lightweight and focused on representing a specific set of concepts. The DL expressivity metric refers to the Description Logics variant adopted by the ontology[25]: *ALU+(D)* indicates a simple ontology (universal restrictions, limited existential quantification) with concept unions (see, e.g., the domain of properties `hasURI` and `hasDataElement` in Subsection 4.1.5) and the use of datatype properties.

Regarding schema metrics:

- *Attribute richness* is the average number of attributes per class and is an indication of both the quality of ontology design and the amount of information pertaining to instance data. The more attributes that are defined, the more knowledge the ontology conveys. The value of 0.818182 demonstrates a high attribute richness for the IDK ontology.
- *Inheritance richness* is the average number of subclasses per class and is a good indicator of how well knowledge is grouped into different categories and subcategories in the ontology. This measure can distinguish a horizontal ontology (where classes have a large number of direct subclasses) from a vertical ontology (where classes have a small number of direct subclasses). The IDK ontology is naturally a vertical ontology, covering a very specific domain in a detailed manner, with a minimal set of subclasses.
- *Relationship richness* is the number of non-inheritance relationships (i.e., object properties, equivalent classes, disjoint classes) divided by the total number of inheritance (i.e., subclass relations) and non-inheritance relationships defined in the ontology. This metric reflects the diversity of the relation types in the ontology, which, in the case of the IDK model, is quite high.
- *Axiom/class ratio* and *class/relation ratio* describe the ratio between axioms-classes and classes-relations, respectively, and are indicators of the ontology's transparency and understandability.

4.3.3 Compliance with User Requirements

As discussed in Subsection 4.1.1, user requirements are mapped to CQs that the ontology is expected to answer. Following the methodology proposed in [21], the CQs were translated into SPARQL queries and evaluated the retrieved results. All of the CQs have been evaluated positively.

4.4 IDK API Specification

The IDK Application Programming Interface (API) provides the means for interacting with the IDK model and the information stored in it. This section presents the first iteration of the IDK implementation by the

[25] https://en.wikipedia.org/wiki/Description_logic

full specification of its API, while the technologies on which the actual implementation of the API will be based on will be decided later according to WP4 and WP5 needs and the outcomes of the pilots during the first iteration of the project.

The first functionality that the IDK has to provide is helping the DE in allowing the user to select the proper cross-border evidence providers, a functionality that has been called “Issuing Authority Location” (IAL). The DE, responsible of the procedure processing, is aware of the evidence that such processing requires and the canonical evidence types that correspond to each required evidence. Thus, for each canonical evidence type, the DE can get the list of its cross-border available sources, a list that can grow over time:

Path	/ial/{canonical evidence type}
Method	GET
Response	List of available provisions of the given canonical evidence type organised by country

Example	/ial/BirthCertificate
	=>
	[
	{
	"countryCode": *"LU"*,
	"atuLevel": *"nuts0"*,
	"provisions": [
	{
	"dataOwnerId": *"iso6523-actorid-upis::9991:LU000000025"*,
	"redirectURL": *"https://ctie.lu/usip"*,
	"dataOwnerPrefLabel": *"CENTRE DES TECHNOLOGIES DE L'INFORMATION DE L'ETAT"*,
	"atuCode": *"LU"*,
	"atuLatinName": *"LUXEMBOURG"*,
	"provisionType": *"usip"*
	}
	]
	},
	{
	"countryCode": *"BE"*,
	"atuLevel": *"nuts1"*,
	"provisions": [
	{
	"dataOwnerId": *"iso6523-actorid-upis::9991:BE000000001"*,
	"redirectURL": *"https:// brussels.gov.be/usip"*,
	"dataOwnerPrefLabel": *"Gouvernement de la Région bruxelloise de Bruxelles-Capitale"*,
	"atuCode": *"BE1"*,
	"atuLatinName": *"Région de Bruxelles-Capitale"*,
	"provisionType": *"usip"*
	},
	{
	"dataOwnerId": *"iso6523-actorid-upis::9991:BE000000002"*,
	"redirectURL": *"https://vlaams.gov.be/usip"*,
	"dataOwnerPrefLabel": *"Regering van het Vlaamse Gewest"*,
	"atuCode": *"BE2"*,
	"atuLatinName": *" Vlaams Gewest"*,
	"provisionType": *"usip"*
	}
	]

```
  },
  {
   "countryCode": "ES",
   "atuLevel": "lau",
   "provisions": {
    "numProvisions": 8123,
    "organisation": "ES/nuts2/nuts3"
  }
]
```


When the list of available issuing authorities within a country is too big, like in the case of Spain (ES) in the previous example, then that list of available provisions can be requested alone:

Path	/ial/{canonical evidence type}/{country code}
Method	GET
Response	List of available provisions of the given canonical evidence type and country

Example	/ial/BirthCertificate/ES
	=>
	{
	"atuLevel": "lau",
	"provisions": [
	{
	"dataOwnerId": "iso6523-actorid-upis::9991:ES000000025",
	"redirectURL": "https://madrid.org.es/usip",
	"dataOwnerPrefLabel": "Ayuntamiento de Madrid",
	"atuCode": "ES28079",
	"atuLatinName": "Madrid",
	"provisionType": "usip"
	},
	{…}
	]
	}

It is also possible to get the characteristics of a particular provision:

Path	/provision?canonicalEvidenceTypeId={}&dataOwnerId={}
Method	GET
Response	Characteristics of the provision corresponding to a given canonical evidence type and data owner
Example	/provision?canonicalEvidenceTypeId=*BirthCertificate*&dataOwnerId=*iso6523-actorid-upis::9991:SI990000105* => { "countryCode": "SI", "atuLevel": "nuts0", "provision": { "dataOwnerId": "iso6523-actorid-upis::9991:SI990000105", "params": [

<table>
<tr><td></td><td><pre>
{
  "title": "SI/nuts3",
  "paramset": [
    "SI/SI031",
    "SI/SI034"
  ]
}
],
"dataOwnerPrefLabel": "Slovenska vlada",
"atuCode": "SI",
"atuLatinName": "SLOVENIJA",
"provisionType": "ip"
}
}
</pre></td></tr>
<tr><td>Example description</td><td>The provision is a IP Proxy provision at national level (nuts0) that requires an input parameter to specify the region (nuts3) to reach the proper issuing authority. Only two regions can provide the evidence, as the “paramset” specifies; if no “paramset” is included, the all the Slovenian regions are available for the provision</td></tr>
</table>

Annex II includes the full specification of the IAL API under the OpenAPI 3.0.3 specification, whose paths have been introduced in this section. This API is implemented by the DE4A Connector developed by the DE4A WP5.

4.5 Summary and Next Steps

This chapter presented the first iteration of the IDK semantic model and the API for accessing the information residing in the IDK knowledge graph, in particular, for helping data evaluators to locate the issuing authorities. The current version of the model will be further refined, along with the API, based on the outcomes from the project pilots in the coming months. The new aspects to consider towards a final version, which is due M24, will include: (a) the addition of a data validation layer on top of the model, via property restrictions (e.g., cardinalities or restrictions to allowed values for properties) that will enable ontological inference, and/or the addition of SHACL shapes [8] for more enhanced data validation; (b) the integration of business rules, like those discussed in Subsection 4.1.3, in the form of SWRL or SHACL rules [22]; (c) preparing the model for publishing on the Web, based on W3C recommendations[26] and related work [23].

[26] Data on the Web Best Practices – W3C Recommendation 31 January 2017: https://www.w3.org/TR/dwbp/

5 Implementation of the Information Exchange Model

The Information Exchange Model (IEM) helps competent authorities to collaborate in the provision of public procedures, by modelling the payload of request and response messages for the evidence exchange. IEM is a common data model for the payload of exchange messages, which models information for processing the request, creating the corresponding response and auditing the correctness of the exchange when this is required. This chapter presents the first version of the IEM, describing the engineering process for designing and developing the model, and also provides a description of the represented information. The chapter concludes with directions towards implementing the final version of the IEM.

5.1 Background

The DE4A project has several precedents such as the PEPPOL[27] and TOOP[28] projects; all three projects exchange information between European parties. PEPPOL was focused on the invoice domain and TOOP covered three pilots on the Public Procurement domain, on Business Registries and on Ship and Crew Certificates. Nevertheless, the TOOP project has been extended to cover the exchange of evidence between public authorities in the processing of administrative procedures, which coincides with the domain of the DE4A scope. Within the same scope, DE4A has also analysed other exchange message models used by national OOP platforms, such as the SCSP protocol of the Spanish Data Intermediation Platform.

Although TOOP and DE4A now share the evidence scope (lawfully issued evidence for processing administrative procedure), their architectures differ in the following significant points, which prevent the use of the TOOP Exchange Data model (EDM) in DE4A:

A. The evidence exchange participant model

TOOP has a two-corner model with two roles involved in the evidence exchange: data consumer (DC) and data provider (DP); SDG names these two parties as Evidence Requestor and Evidence Provider. DEA4 has a four-corner model with four roles involved: two from the business point of view –data evaluator (DE) and data owner (DO), who are the competent authorities for the procedure and the evidence respectively– and two from the technical point of view –data requestor (DR) and data transferor (DT), who are the technical parties for mentioned competent authorities respectively; nevertheless, the same organisation can play both the business and technical role at the requestor side or at the provider side.

Actually, TOOP is also using a four-corner model because this is the model used by the CEF eDelivery, i.e., business and technical parties in both sides of the iteration. However, TOOP considers both business and technical parties of each side as a whole (DC+DP) whilst DE4A considers the four parties (DE/DR+DT/DO).

B. The evidence request model

TOOP's evidence request model is based on two alternatives that have been considered in the TOOP EDM design:

1. **Concept-Query (criteria-based approach)**: Request for some atomic concepts that are defined in the agreed TOOP ontology. The response contains the values for the requested concepts in the case of the specified data subject.
2. **Document-Query (document-based approach)**: Request for some evidence type, identified by a dataset identifier, obtains as response document metadata (**two-step** approach) or attached

[27] https://peppol.eu/

[28] https://www.toop.eu/

documents in any format (**one-step** approach). This query may ask for a particular document format or, after receiving the document metadata, for a document ID. This functionality is of interest in public procurement, because the existence of the evidence is the only parameter required to present offers; the document is only required to sign the contract.

The DE4A evidence request model is based on a **canonical-based approach** and the DE4A IEM has been designed in consequence.

According to Deloitte's study[29] and the methodology proposed by the SDG OOP WP4 on Semantics, DE4A pilots produce a list of canonical evidence types with agreed common data models, which have been implemented as XML schemas. These agreed schemas allow the automatic processing of canonical evidence regardless of the issuing authority. However, when the evidence legal value only relies on the domestic evidence that is originally issued by the corresponding authority, this original domestic evidence needs to be attached to the canonical evidence for legal validity. Besides, some public documents have a multilingual form according to Regulation (EU) 2016/1191[30], so the issuing authority could also want to attach the multilingual form document[31] that includes specific-country information. Domestic original evidence and multi-lingual forms may be issued by the issuing authority in any mime-type. All of these evidence instances are relevant because future possible audits of the procedure processing might require reviewing the correspondence between the contents of the evidence with legal value and the contents of the canonical evidence.

C. The request parameters in the Intermediation Pattern

TOOP implements a direct intermediation pattern, which only models interactions between a user and a data consumer, and between a data consumer and a data provider for getting the required evidence. According to this pattern and the TOOP EDM[32], the requested evidence has to be located from the user's eIDAS dataset exclusively. However, some evidence types need more information to locate the proper evidence data; for instance, the proof of a vehicle registration could require the plate number along with the owner's eIDAS dataset. In this regard, DE4A has considered the case of including in the request message the additional parameters that an evidence provision can require to properly locate the requested evidence, i.e., for the record matching.

DE4A also implements a user-supported intermediation pattern that includes a direct interaction between the user and the data owner to formalise the evidence request. In this case, any additional information required by the issuing authority to locate the proper evidence will be asked to the user at the data owner's portal, so the request message does not need to include it.

D. Representation of data subjects

TOOP does not consider the representation of natural persons, whilst DE4A has considered in the request the case of natural persons representing data subjects, either natural or legal persons. Consequently, the DE4A IEM has been designed to cover such a case.

[29] Study on Data Mapping for the cross-border application of the Once-Only technical system SDG. 28/02/2020.

[30] Regulation (EU) 2016/1191 of the European Parliament and of the Council of 6 July 2016 on promoting the free movement of citizens by simplifying the requirements for presenting certain public documents in the European Union and amending Regulation (EU) No 1024/2012.

[31] The ISA[2] team implemented these multilingual forms with XML schemas.

[32] TOOP Exchange Data Model Specification v1.4.1 Data Model Request at http://wiki.ds.unipi.gr/display/CCTF/TOOP+Exchange+Data+Model+Specification+v1.4.1#TOOPExchangeDataModelSpecificationv1.4.1-DataModel:TOOPRequest

5.2 Development of the Model

The development of IEM has been based on the requirements provided by the DE4A architecture, pilots and technical design of common components. Besides, the IEM development has used some relevant existing semantic assets.

5.2.1 Requirements Elicitation

The DE4A IEM follows the basic assumptions and requirements included in the following two tables:

Table 14: DE4A IEM basic assumptions.

The DE4A IEM allows the message exchange between data owners and data evaluators for the processing of administrative procedures.
A data owner, along with its data transferor, can automatically generate an IEM response message for the received request message.
The DE4A IEM models common information to include as payload of request and response messages, including error responses.
The IEM is abstract enough to allow the exchange of data or documents in any business domain.
The DE4A IEM is based on the agreed canonical forms of evidence types and existing European or international vocabularies or standards.
The IEM satisfies the specific needs of the DE4A architecture, pilots and technical common components.

Table 15: DE4A IEM basic requirements.

The DE must be able to request from the DO information about the user of the procedure, to be used as evidence in the procedure processing.
The DE must provide data that identifies the user, and the data subject if they are not the same one.
The DC must specify that the user explicitly requested the use of the Once-Only technical system (OOTS) for the retrieval of the evidence or to state that a law prevents such an explicit request.
The DE must be able to specify the purpose of the evidence usage by providing the name of the involved procedure.
Competent authorities involved in the evidence exchange must be identified.
The DT must be able to transmit the requested information to the DR.
A response should unambiguously refer to its corresponding request.
The DE must be able to unambiguously understand and automatically process evidence by its canonical form.
The DO must be able to provide evidence with legal value in a format lawfully issued.
The DO must be able to provide evidence in the public documents domain according to its multilingual form in a format selected by the authorities.
If the DP cannot transmit the requested evidence, the reasons must be given.
Transmitted evidence shall be limited to what has been requested according to the agreed canonical evidence types.

For outlining the scope of information to include in the IEM, and following a similar requirements elicitation procedure like the one described in the previous chapter, a set of **Competency Questions (CQs)** is used, namely, natural language queries that express a pattern for a type of question the IEM should be

able to answer. In essence, CQs represent functional requirements that the IEM should be able to respond to. Thus, CQs are used for the specification, as well as for the validation of the IEM and for the possible audits of the exchange. In this context, the following tables show the CQs for the request and response messages in the IEM.

Table 16: Competency Questions for the Request IEM.

CQ#	Competency Question	Answer and Sample Response
CQ1	What is the format of the request message [R]?	The corresponding to the Specification identified in the message.
CQ2	How is a request message [R] identified?	By a UDDI.
CQ3	When was request [X] sent?	At the specific TimeStamp.
CQ4	What canonical evidence type is requested in a request message [R]?	The corresponding to the canonical evidence type id included in the message.
CQ5	How is canonical evidence type [X] identified?	With a URN, according to the DE4A policy for identifiers, like, e.g., `urn:eu-de4a:CanonicalEvidenceType::HigherEdCertificate:1.0`.
CQ6	What authority is requesting the evidence in the request message [R]?	The data evaluator [Y], who is responsible for processing the evidence.
CQ7	What authority is requested to issue the evidence in the request message [R]?	The data owner [Z], who is responsible for lawfully issuing the requested evidence.
CQ8	How is a data evaluator [Y] or a data owner [Z] identified?	With a URN according to the ISO 6523 specification, i.e., `iso6523-actorid-upis::9991:PT990000101`, and optionally with one or more preferred names.
CQ9	What is the purpose of the evidence request [R]?	Considering the evidence data in the processing of an administrative procedure [X].
CQ10	How is the procedure [X] identified?	If the procedure requesting the evidence is identified, the procedure designation is included in the message.
CQ11	Who is the subject of the requested evidence [X] about?	The data subject.
CQ12	Is the data subject [X] a natural or a legal person?	Either type of person.
CQ13	Is the data subject the user of the procedure?	Yes, except if the user is a natural person representing the data subject (e.g., father requesting his child's birth certificate).
CQ14	How is the data subject [X] identified?	With the mandatory dataset established by the eIDAS regulation.
CQ15	How is the data subject's representative [X] identified?	With the mandatory dataset established for natural persons by the eIDAS regulation, along with any additional parameter required by the evidence provider.

Table 17 shows the CQs that drive the design of the model for representing the response messages in the IEM.

Table 17: Competency Questions for the Response IEM.

CQ#	Competency Question	Answer and Sample Response
CQ1	What is the format of the response message [R]?	The corresponding to the Specification identified in the message.
CQ2	What request message is the response message [R] responding to?	The request message identified by the specified Request Id.
CQ3	When was response [X] sent?	At the specific TimeStamp.
CQ4	What authority is requesting the evidence in the request message [R]?	The data evaluator [Y], who is responsible for processing the procedure.
CQ5	What authority is requested to issue the evidence in the request message [R]?	The data owner [Z], who is responsible for lawfully issuing the requested evidence.
CQ6	How is a data evaluator [Y] or a data owner [Z] identified in the message?	With a URN according to the ISO 6523 specification, i.e., `iso6523-actorid-upis::9991:PT990000101`, and optionally with one or more preferred names.
CQ7	Who is the person subject of the requested evidence that response message [R] is responding to?	The data subject as it was identified in the corresponding request message.
CQ8	What is the response provided in the response message [R]?	Either an error or the evidence requested in the corresponding request message.
CQ9	What error information is provided in the response message [R]?	The code, the message and, optionally, any other relevant additional information of the error (e.g., 1403, “Evidence not found”, “No record found for the given and family name of the data subject”).
CQ10	What information is provided in response message [R] about the requested evidence?	The requested evidence in its canonical format and, optionally, if the canonical form has not legal value, in its domestic original format and/or its multilingual format with legal value.
CQ11	How is a canonical format of an evidence [X] represented?	With a canonical evidence type URN, according to the DE4A policy for identifiers, like, e.g., `urn:eu-de4a:identifiers:CanonicalEvidenceType::HigherEdCertificate:1.0`, along with the data evidence in according to the agreed XML Schema for such canonical evidence type.
CQ12	How is an original domestic format of an evidence [X] represented?	With any format represented in base64 encoding according to the specified mime-type (e.g., application/pdf), data language (iso 639-2 code, e.g., “es”) and, optionally, any other additional information (e.g., “scanned original page of the birth civil registry volume”).
CQ13	How is a multilingual form format of an evidence [X] represented?	With any format represented in base64 encoding according to the specified mime-type (e.g., application/pdf), data languages (iso 639-2 codes for the original language and other language, e.g., “es/en”) and, optionally, any other additional information (e.g. “multilingual form of birth certificate according to Regulation (EU) 2016/1191”).

5.2.2 Reusing Third-party Resources

A typical step in semantic engineering involves considering the reuse of existing third-party resources. The IEM implementation has been inspired by TOOP EDM, and it also uses the following third-party resources:

- eIDAS SAM Attribute Profile XML Schemas (v1.2): These schemas (`http://eidas.europa.eu/attributes/legalperson`, `http://eidas.europa.eu/attributes/naturalperson`) were developed by the eIDAS eID Technical Subgroup and were adopted by the eIDAS Cooperation Network. These schemas have been used to represent the data subject, natural or legal person, and the user's request in the case where the latter is representing the former.
- SEMIC Common data types XML Schema: This schema (`https://semic.org/sa/cv/common/dataTypes-2.0.0`) was used by ISA[2] when they developed the XML Schema for the multilingual forms of public documents. This schema uses the module of Core Component Type developed by UN/CEFACT (`urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2`).

5.2.3 Specifying Key Concepts

Following the same methodology with the one presented in the previous chapter, the key concepts to be included in the IDK model have been obtained from the requirements established by the DE4A project architecture, design and pilot outcomes. The following are the core concepts for the IEM, along with a respective set of "business rules", which are aligned with the key concepts previously described in Section 4.1.3. Core concepts are annotated in **bold dark blue font**:

IEM Request: An IEM request is the payload of a message sent by a data evaluator to a data owner to obtain evidence of a certain type about a subject:

- An IEM request is a set of data elements according to a certain **specification** of the IEM.
- An IEM request happens at a certain **timestamp**.
- An IEM request is identified by a **Universally Unique Identifier (UDDI)**.
- An IEM request could specify the **procedure** that will process the evidence by its name or any other text, according to the procedure legal rules.
- An IEM request identifies the **grounds** of the use of the OOTS, either a user's explicit request or a law that avoids it.
- A **data evaluator** is the competent authority for processing the requested evidence.
- Data evaluators request evidence according to a **canonical evidence type** that is relevant for processing an administrative procedure.
- A **data evaluator** is identified with a **URN** according to the ISO 6523 specification.
- A **data subject** is the subject about evidence required, either a natural or a legal person; a natural person might be represented by another natural person that is the user of the procedure, when the procedure rules allow such a representation (e.g., a father that requests his child's birth certificate). A data subject is identified by the dataset defined by the eIDAS Regulation.
- If necessary, a request includes the value of a set of additional parameters needed to properly locate the requested evidence besides the data subject identification (record matching).

IEM Response: An IEM response is the payload of a message sent by a data owner to a data evaluator with either the requested evidence or the description of the error that prevented the evidence delivery.

- An IEM response is a set of data elements according to a certain **specification** of the IEM.
- An IEM response happens at a certain **timestamp**.
- An IEM response identifies the corresponding request by its **Universally Unique Identifier (UDDI)**.
- A **data owner** is the competent authority for lawfully issuing evidence of a certain type.
- Data owners deliver evidence according to a **canonical evidence type** that is lawfully required for processing a procedure of the data evaluator's competence.
- A **data owner** is identified with a **URN** according to the ISO 6523 specification.

- An IEM response returns either the required evidence or the error that prevents the evidence delivery.
- The error is described by a **code**, an error **message** and, optionally, **additional information**.
- The returned evidence is the **canonical evidence** optionally along with the **domestic evidence**.

Canonical Evidence: a canonical evidence provides information about a **data subject** –natural or legal person– according to an agreed common data model for the corresponding canonical evidence type.

- A **canonical evidence type** is identified with a **URN** according to the DE4A policy for identifiers.
- A canonical evidence is composed of **elements** according to the XML Schema for the common data model of the corresponding canonical evidence type. This common data model allows the automatic processing of the evidence according to the rules of the corresponding administrative procedure, regardless the issuing authority.

Domestic Evidence: a domestic evidence provides information related to **canonical evidence type** regarding a **data subject** and according to the format with legal value originally issued by the issuing authority and, optionally in the case of public documents, the multilingual form of such original evidence.

- The domestic evidence **format** is according to its mime-type.
- A domestic evidence **issuing type** might be either an originally issued evidence or, in the case of a public document, a multilingual form.
- A domestic evidence contains information in one or several **languages**.
- A domestic evidence might include some text with **additional information**.
- A domestic evidence is encoded and attached in the response message.
- A domestic evidence with legal value will be required in audits of either the procedure processing or the evidence exchange, to prove the equivalence between the information provided by such domestic evidence and the corresponding canonical evidence.

Data subject: processing of a procedure might require evidence on some fact regarding a subject, that is the data subject of the evidence exchange:

- A data subject might be a **natural or a legal person**.
- A data subject is normally the user of the procedure with one exception according to the DE4A Moving Abroad pilot needs: when the user is a natural person, and the procedure requires evidence regarding another natural person –e.g., the user is a father, and the required evidence is his child's certificate– then the user is the **data subject representative**.
- The information available about the identity of the user of the procedure is based on the eIDAS dataset.
- In the case of a represented data subject, its identity information is according to the eIDAS dataset for natural persons.

5.3 IEM Overview

Public authorities responsible for administrative procedures obtain pieces of evidence from public authorities that lawfully issue them by means of IEM request and response messages. In consequence, the IEM has to identify the agents, evidence, data subject and messages involved in each exchange, in order to properly locate the required evidence and to allow any audit on the evidence exchange that might be required in the future. The IEM is represented with an XML Schema to be included as the payload of the request and response messages by the WP5 Common Component Design and Development, according to the technical specifications.

5.3.1 IEM Request

By means of IEM request messages, public authorities responsible of administrative procedures request evidence to public authorities that lawfully issue such evidence. For this purpose, IEM needs to identify the request message, the involved competent authorities, the data subject and the requested evidence (see Figure 19).

The ***request exchange*** is identified by a Universally Unique Identifier (UDDI) of the request message (`RequestId`), the identifier of the IEM version (`SpecificationId`), the timpestamp of the request (`TimeStamp`), the ground of the request through the OOTS (`RequestGrouds`) –either a user's explicit request or a legal ground– and, optionally, the procedure that requires the evidence (`Procedure`). In this initial version, the element field is an optional free text with the procedure name; this element type will be reviewed in a latter phase of the project, when the issue of access authorization to required evidence will be tackled.

The ***involved competent authorities*** are (a) the public organization responsible for processing the evidence in the context of an administrative procedure (`DataEvaluator`), and (b) the public organization that lawfully issues the requested evidence (`DataOwner`). Both public organizations are agents identified by a URN (`AgentUrn`) according to the ISO 6523 specification and, optionally, one or more names as preferred labels of the public organizations (`AgentName`).

The ***data subject*** (`DataRequestSubject`) might be either a natural person (`DataSubjectPerson`) or a legal person (`DataSubjectCompany`) according to the eIDAS dataset specification. In both cases, the data subject might be represented by a representative (`DataSubjectRepresentative`); in this case, the representative is the user that has been identified at the procedure portal. The DE4A pilots have limited the scope of the representation to a natural person representing another natural person when, for instance, a father requests his child's birth certificate.

The ***requested evidence*** is identified with the corresponding canonical evidence type URN (`CanonicalEvidenceTypeId`) and the URN of the data owner (`DataOwner`), along with the input parameters that the evidence provisioin might require besides the data subject identification dataset (`AdditionalParameters`). Each input parameter (`AdditionalParameterValue`) is represented by a value (`TermValueType>TermValue`) and the URI path of a term registered in the IDK (`TermValueType>TermMorUri`), where the type and the multilingual label and description of the term can be found.

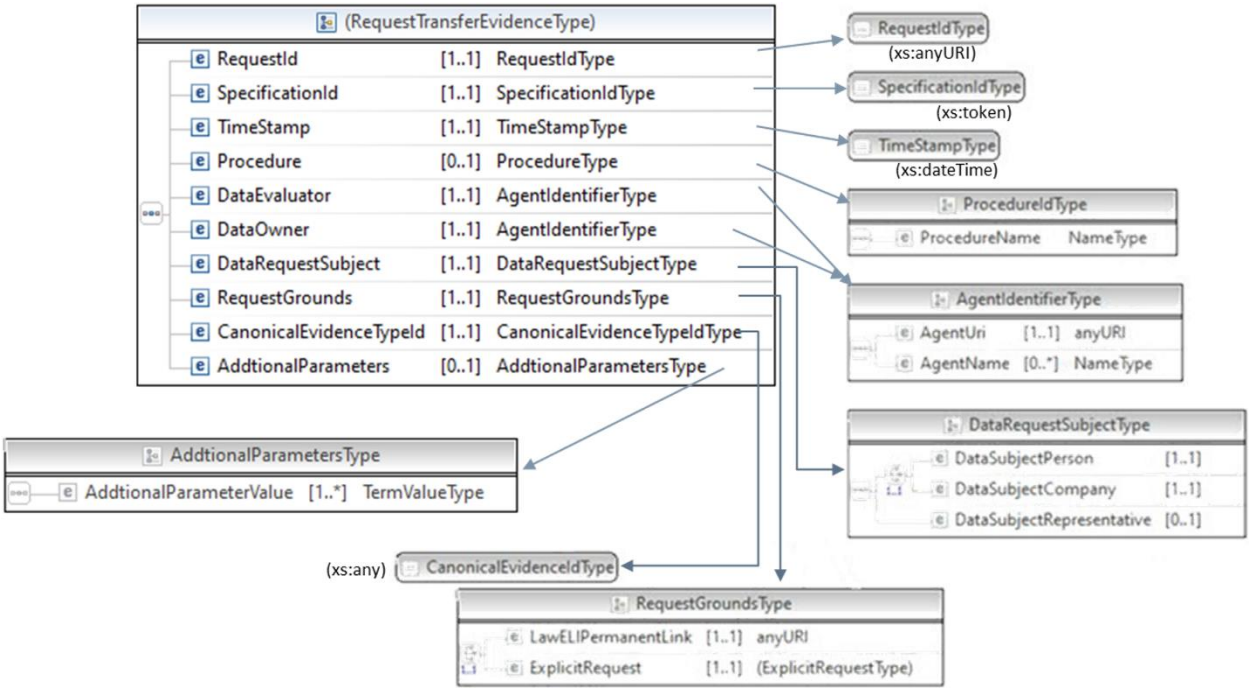


Figure 19: IEM Request XSD representation.

5.3.2 IEM Response

By means of IEM response messages, public authorities provide lawfully issued evidence to public authorities responsible for processing administrative procedures that require that evidence. For this purpose, IEM response messages identify the corresponding request message, the involved competent

authorities and the data subject, and they also include the response to the request, i.e., either an error or the requested evidence in both their canonical and domestic forms, the former for semantic interoperability and the latter for legal interoperability Figure 20).

The UDDI of the **request message** (`RequestId`) is used to link the response to the corresponding request message. However, ***involved competent authorities*** (`DataOwner` and `DataEvaluator`) and ***data subject*** (`DataRequestSubject`) are included to provide redundant information to verify the link between response and request. The request message, involved competent authorities, data subject, model specification version and time are modelled in the same way as in the IEM Request. These elements, along with the ***timestamp*** of the response message, also provide the metadata of the evidence issuing, and the ***response*** element (`Response`) contains either the issued evidence or the error that prevents the former. With the exception of the response element, the rest of the elements are modelled in the same way as in the IEM Request, including the identifier of the specification of the IEM schema.

As mentioned above, the *response* could be either an error or the required evidence. In the first case, the ***error response*** (`ErrorResponse`) includes the error code (`ErrorResponse>Code`), the error message (`ErrorResponse>Message`) and, optionally, other relevant text (`ErrorResponse>AdditionalInfo`). In the second case, the ***successful response*** (`EvidenceResponse`) includes the canonical evidence, which provides semantic interoperability and optionally, if the canonical evidence is not entitled to have legal value, one or more domestic evidences that provide such legal value.

The ***canonical evidence*** element (`EvidenceResponse>CanonicalEvidence`) includes the URN of the corresponding canonical evidence type (`EvidenceResponse>CanonicalEvidence>CanonicalEvidenceTypeId`) and its data according to its common data model (`EvidenceResponse>CanonicalEvidence>CanonicalEvidenceData`). Each ***domestic evidence*** (`EvidenceResponse>DomesticEvidence`) includes a reference to the attached document (`EvidenceResponse>DomesticEvidence>DomesticEvidenceRef`), its mime type (`EvidenceResponse>DomesticEvidence>MimeType`), the type of issuing –original evidence or multilingual form evidence– (`EvidenceResponse>DomesticEvidence>IssuingType`), the language(s) of its contents (`EvidenceResponse>DomesticEvidence>DataLanguage`) and, optionally, other relevant information (`EvidenceResponse>DomesticEvidence>AddtionalInfo`).

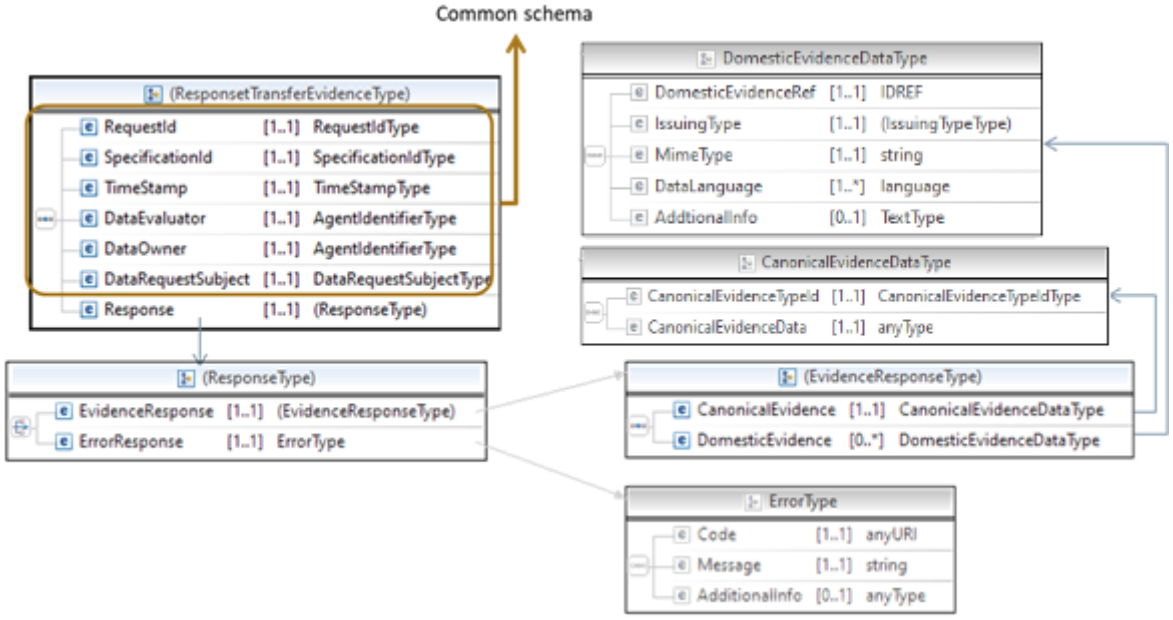


Figure 20: IEM Response XSD representation.

5.4 Summary and next Steps

The DE4A IEM helps competent authorities to collaborate in the provision of public procedures by modelling the payload of request and response messages for the evidence exchange. The payload of the response messages contains both the metadata of the evidence issuing and the evidence in its canonical and domestic forms that provide semantic and legal interoperability in a cross-border scenario.

The DE4A IEM has been designed accordingly with the DE4A architecture and pilot needs, in close collaboration with DE4A WP5 Common Component Design. DE4A WP5 is implementing interfaces for authorities involved in the exchange to make the composition of request and response messages in general and the IEM in particular more transparent.

At the first iteration of the project, the running DE4A pilots will test the properness and validity of the IEM design. Any error or improvement identified during the first iteration regarding the IEM will be considered at the next iteration of the project. The IEM initial version also considers functionalities that will not be tested during the project's first iteration, such us the procedure identification for authorization purposes or the use of additional input parameters required by an evidence service to properly locate the requested evidence (record matching). The inclusion of such functionalities is for covering all the elicited requirements and keeping them visible to all from the beginning.

6 Conclusions

This deliverable introduces an initial version of the semantic layer stack toolkit for delivering cross-border public services related to the DE4A pilot use cases. A comparative study on existing Semantic Web technologies and tools for knowledge representation and semantic modelling is conducted. More specifically, the deliverable provides an overview of the existing tools for creation of XSD files ontology management and their storage, according to DE4A-related requirements. This process helped DE4A to select the right tools that will be part of the semantic toolkit. Furthermore, XSD files are implemented for the canonical evidence types that will cover the needs of the MVP by using the minimum set of common attributes provided by all pilot MS. Aiming to fulfill the MVP requirements, it is also described the implementation process of the Information Desk's basic elements that will facilitate DCs and DPs to obtain the required information before making requests and/or sending responses to the respective stakeholders. This includes the ontology development and overview using competency questions, the ontology evaluation (in terms of consistency, quality, structure and compliance with the requirements) and the IDK RESTful API for interacting with the IDK model and the information stored in it. In the same manner, the deliverable describes the implementation process for the basic design of IEM for modeling the payload of request and response messages for the evidence exchange.

This deliverable presents the semantic components developed for the minimum viable product (MVP) as at the time of the submission, a.k.a February 2021. This version of the semantic interoperability framework and the tools are lightweight; i.e., they are tailored for the simplest scenarios defined by the three DE4A pilots (SA, DBA, MA) teams. Major simplifications include that the canonical evidences carry hard coded data that resonates the real examples rather than real data retrieved in real time from live environments. Consequently, a simpler version of the IDK that supports the handling of such mimicked data as well. Furthermore, the full power of the existing vocabularies and the code lists are not yet exploited in developing the canonical evidences. The main reasoning behind ranking this fact at a lower level of priority is that, more attention is paid to the possibilities of member states to map the canonical evidences to their respective domestic evidences enabling the automatic exchange of evidences across borders. Although this mapping is done partially by the member states involved in the first iteration of piloting, during the validation stage of the canonical evidences (Figure 9), it is of a high relevance to investigate how far the domestic data can be harmonized with a common data model (XML schema), during execution of cross-border procedures, hence full stack evidence-based approach could be demonstrated.

During the first iteration, the suitability and validity of the semantic components (canonical evidence types, IDK, IEM) will be tested in a few selected member states. As a result of piloting, it is expected to systematically capture the quality of the cross-border evidence exchange. Not only the successful attempts, but also the errors and failures are summarized into an issues log, allowing to make decisions on possible improvements to the meta data models of the Once -Only Technical System (OOTS) as well as associated semantic assets including canonical evidences, based on the analysis of the issues logs. Such improvements and modifications to the semantic components are intended to be piloted and verified during the second iteration towards the end of the project.

Moreover, future work includes the detailed specification of the canonical evidence types with more complex data types (with more semantic mappings), redefined cardinalities and code lists that will facilitate all the requirements of the next iteration. Based on the outcomes of the first iteration, it is anticipated to sought out solutions for the failures in automated exchange of evidences across borders using more advanced technologies from cryptography or machine learning domains.

The IDK at its current version does not showcase its complete functionality, that includes registries for locating the authority that lawfully issue evidences, the locater of relevant domestic evidence, the multilingual ontology repository for translating the attributes to domestic languages and the registry for cross-border access authorization. It is assumed that the outcomes of the first iteration may set the grounds for deciding on how and in which ways the IDK could be validated and extended with these business rules.

References

[1] Extensible Markup Language (XML) 1.0 (5th Ed.) – W3C Recommendation 26 November 2008: https://www.w3.org/TR/xml/

[2] W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures – W3C Recommendation 5 April 2012: https://www.w3.org/TR/xmlschema11-1/

[3] W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes – W3C Recommendation 5 April 2012: https://www.w3.org/TR/xmlschema11-2/

[4] Gruber, T. (2009). Ontology. Encyclopedia of database systems, 1, 1963-1965.

[5] RDF 1.1 Primer – W3C Working Group Note 24 June 2014: https://www.w3.org/TR/rdf11-primer/

[6] RDF Schema 1.1 – W3C Recommendation 25 February 2014: https://www.w3.org/TR/rdf-schema/

[7] OWL 2 Web Ontology Language Primer (2nd Edition) – W3C Recommendation 11 December 2012: https://www.w3.org/TR/owl2-primer/

[8] Shapes Constraint Language (SHACL) – W3C Recommendation 20 July 2017: https://www.w3.org/TR/shacl/

[9] Bizer, C., Heath, T., & Berners-Lee, T. (2011). Linked data: The story so far. In Semantic services, interoperability and web applications: emerging concepts (pp. 205-227). IGI global.

[10] Armandoa et al., VocBench 3: A Collaborative Semantic Web Editor for Ontologies, Thesauri and Lexicons Semantic Web 11(5), 855-881, 2020.

[11] Stellato, A. et al., VocBench: A Web Application for Collaborative Development of Multilingual Thesauri. Proc 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, 2015, pp 38-53.

[12] SPARQL 1.1 Query Language – W3C Recommendation 21 March 2013: https://www.w3.org/TR/sparql11-query/

[13] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). SWRL: A semantic web rule language combining OWL and RuleML. W3C Member submission, 21(79), 1-31

[14] Weichbroth, P. (2019). Fluent Editor and Controlled Natural Language in Ontology Development. International Journal on Artificial Intelligence Tools, 28(04), 1940007. doi:10.1142/s0218213019400074

[15] Atemezing, A. G., and Amardeilh, F., Benchmarking Commercial RDF Stores with Publications Office Dataset, Proceeding of ESWC 2018: The Semantic Web: ESWC 2018 Satellite Events pp 379-394, 2018

[16] N.F. Noy and D.L. McGuinness. Ontology Development 101: A Guide to Creating your First Ontology. Technical Report KSL-01-05, and Stanford Medical Informatics Technical Report SMI-2001-0880, Stanford Knowledge Systems Laboratory, March 2001.

[17] M. Uschold, M. Gruninger, Ontologies: Principles, methods and applications, Knowledge Eng. Review 11 (2) (1996) 93–136. doi:10.1017/S0269888900007797

[18] Miles, A., & Bechhofer, S. (2009). SKOS simple knowledge organization system reference. W3C recommendation.

[19] Falco, R., Gangemi, A., Peroni, S., Shotton, D., & Vitali, F. (2014, May). Modelling OWL ontologies with Graffoo. In European Semantic Web Conference (pp. 320-325). Springer, Cham.

[20] Poveda-Villalón, M., Gómez-Pérez, A., & Suárez-Figueroa, M. C. (2014). Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation. International Journal on Semantic Web and Information Systems (IJSWIS), 10(2), 7-34.

[21] Zemmouchi-Ghomari, L., & Ghomari, A. R. (2013). Translating natural language competency questions into SPARQL Queries: a case study. In 1st Int. Conf. on Building and Exploring Web Based Environments, pp. 81-86.

[22] SHACL Advanced Features – W3C Working Group Note 08 June 2017: https://www.w3.org/TR/shacl-af/

[23] Gyrard, A., Serrano, M., & Atemezing, G. A. (2015, December). Semantic web methodologies, best practices and ontology engineering applied to Internet of Things. In 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT) (pp. 412-417). IEEE.

[24] SDGR, 14401/1/17 REV 1, Brussels, 28 November 2017, Article 3(1)

[25] Ontologies - W3C. (2020). Retrieved 29 September 2020, from https://www.w3.org/standards/semanticweb/ontology

[26] EUR-Lex — Access to European Union law — choose your language. (2020). Retrieved 29 September 2020, from https://eur-lex.europa.eu/

[27] LibGuides. (2020). Retrieved 29 September 2020, from https://pitt.libguides.com/

[28] Frade, J., 2021. *Once Only Initiative: Towards Cross-Border Connectivity*. [online] CEF Digital. Available at: <https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Once+Only+Initiative%3A+Towards+Cross-Border+Connectivity> [Accessed 17 February 2021].

Annexes

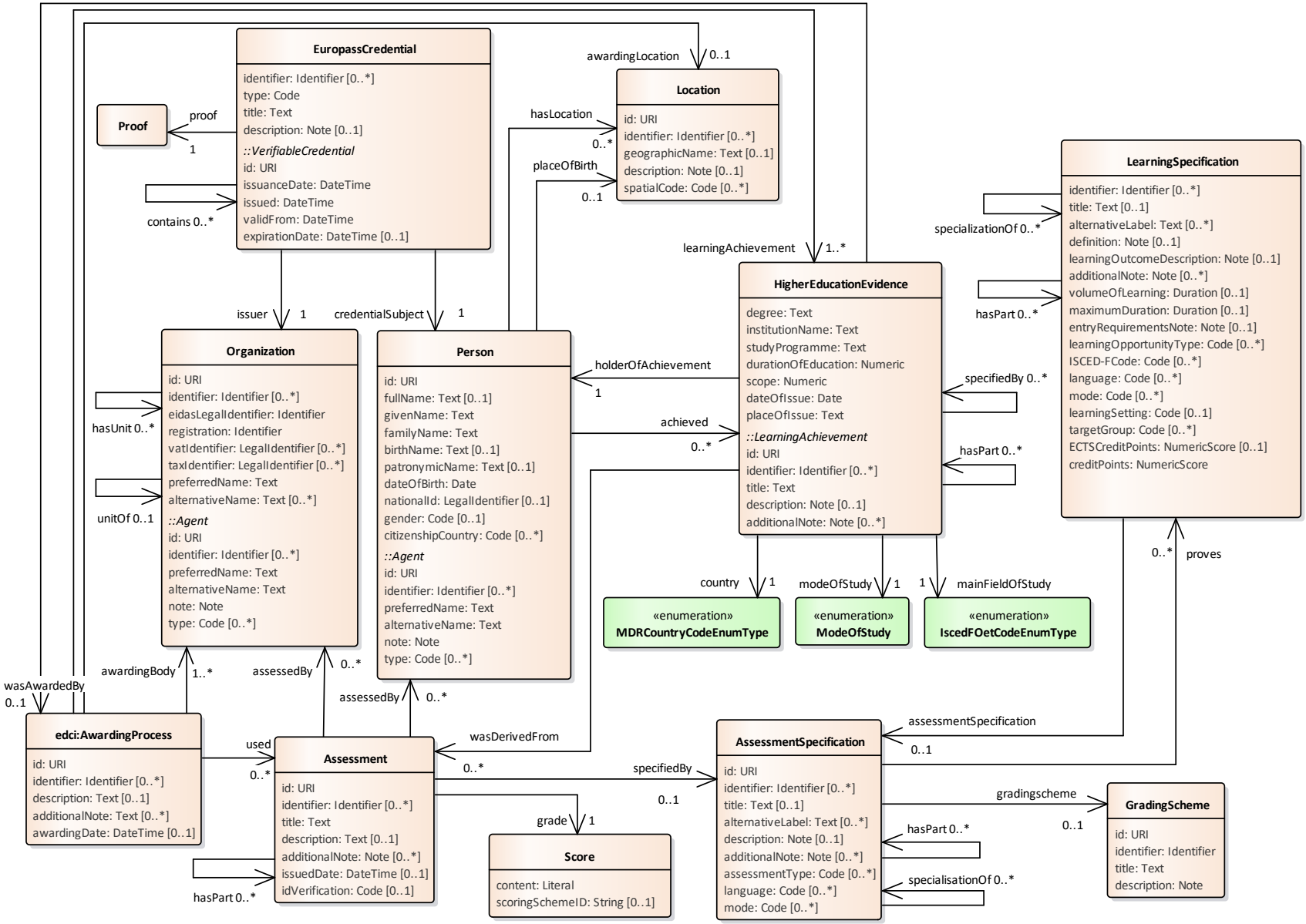
Annex I. Studying Abroad Application Profile

The following classes and properties is a subset of the classes and properties of the Europass Learning Model as it was documented on GitHub as of January 31 2021. It is proposed that these classes and properties should be used as the core model for the Studying Abroad pilot. In addition to the shown EDCI-defined model elements, it is also proposed that subclasses of '`LearningAchievement`' will be defined for '`HigherEducationEvidence`' and '`SecondaryEvidence`'. It has to be decided if these two classes should be defined within the Europass namespace or an alternative namespace.

- Europass Learning Model:

  https://github.com/european-commission-europass/Europass-Learning-Model/blob/master/Credentials/Credentials_Learning_Model.md

Overview of the proposed Studying Abroad Application Profile



Annex I-a. Classes and properties defined by DE4A

HigherEducationEvidence

Class description: The diploma data DCs need from a student.

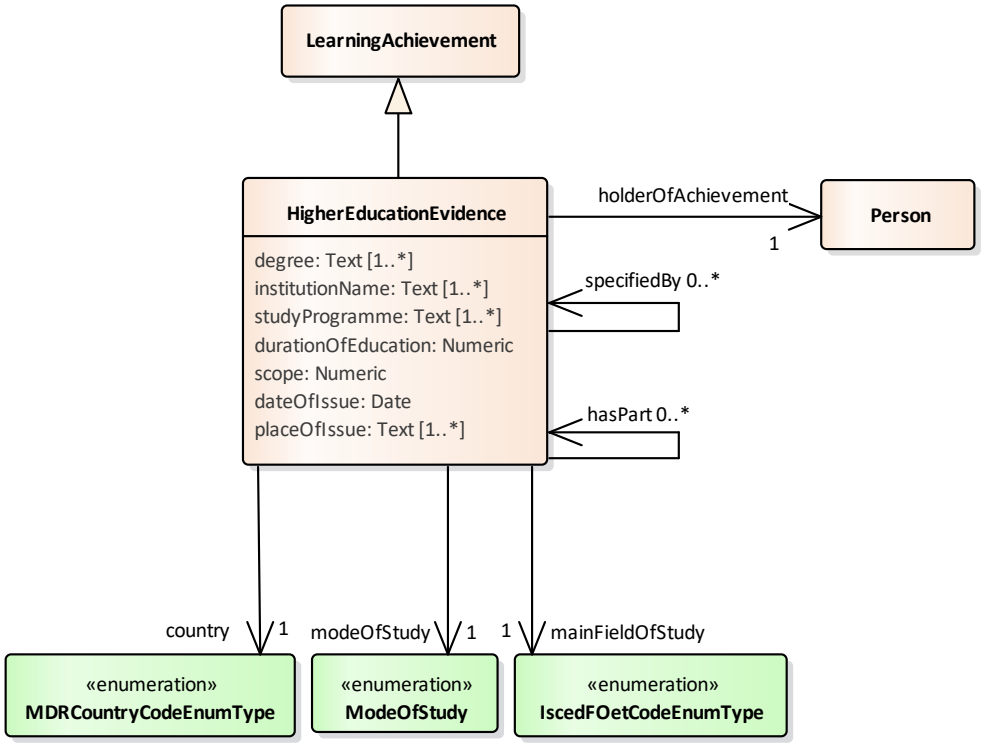


Table 18:HigherEducationEvidence detailed class – definitions, data types and cardinalities

Label	Definition	Field	Range (data type)	Card
degree	An academic title or degree obtained by the student and proven by this diploma or certificate (evidence)	degree	Text	1 - *
institution name	The name of the higher education institution where the student obtained the degree	institutionName	Text	1 - *
study programme	Name of a study programme that the student finished at the higher education institution in order to obtain the degree	studyProgramme	Text	1 - *
duration of education	Official duration of education in years	durationOfEducation	Numeric	1
scope	The official workload of the study programme	scope	Numeric	1

Label	Definition	Field	Range (data type)	Card
	in the ECTS (European Credit Transfer and Accumulation System) credit points			
date of issue	Date of issue of the certificate or diploma	dateOfIssue	Date	1
place of issue	Place of issue (location) of the certificate or diploma	placeOfIssue	Text	1 - *
country	Country where the study programme was completed by the student	country	MDRCountryCodeEnumType	1
main field of study	Field of finished higher education	mainFieldOfStudy	IscedFOetCodeEnumType	1
mode of study	Mode of study, (full time, part time, distance learning)	modeOfStudy	ModeOfStudy	1

ModeOfStudy (enumeration)

Class description: Enumeration class for the mode of study, Contains three enumerations: 'full time', 'part time' and 'distance learning'.

Table 19:ModeOfStudy detailed class – definitions, data types and cardinalities

Namel	type
distance learning	xsd:string
full time	xsd:string
part time	xsd:string

Annex I-b. Classes and properties defined be EDCI

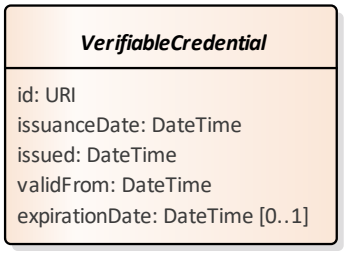
The following are the subset of EDCI classes and properties that is used in the Studying Abroad application profile. All properties and association is documented as describe in the EDCI documentation.

The description of classes matches that of the EDCI documentation.

If the range of an association (object property) is a class that is not included in the application profile, the class is only given a short description without the details of any properties.

Verifiable Credential < abstract >

Class description: A set of one or more claims made by an issuer. A credential is a set of one or more claims made by the same entity. A verifiable credential is a tamper-evident credential that has authorship that can be cryptographically verified. Verifiable credentials can be used to build verifiable presentations, which can also be cryptographically verified.



[Note that this is an EDCI representation of the class'VerifiableCredential', part of the W3C Verifiable Credentials Data Model'[33].]

Table 20: VerifiableCredential detailed class – definitions, data types and cardinalities

Label	Definition	Field	Type Range (data type)	Card
Credential UID	A unique portable identifier of the credential.	id	ID/PK URI	1
Issuance Date	The issuance date	issuanceDate	Property DateTime	1
Issue Date	The date and time the credential was digitally signed.	issued	Property DateTime	1
Valid From	The earliest date when the information associated with the credentialSubject property became valid	validFrom	Property DateTime	1
Expiry Date	The expiration date.	expirationDate	Property DateTime	0. .1

[33] https://www.w3.org/TR/vc-data-model/#dfn-verifiable-credentials

Europass Credential < extends VerifiableCredential >

Class description: A set of claims made by an issuer in Europe, using the Europass Standards. A Europass credential is a set of one or more claims which may be used to demonstrate that the owner has certain skills or has achieved certain learning outcomes through formal, non-formal or informal learning.

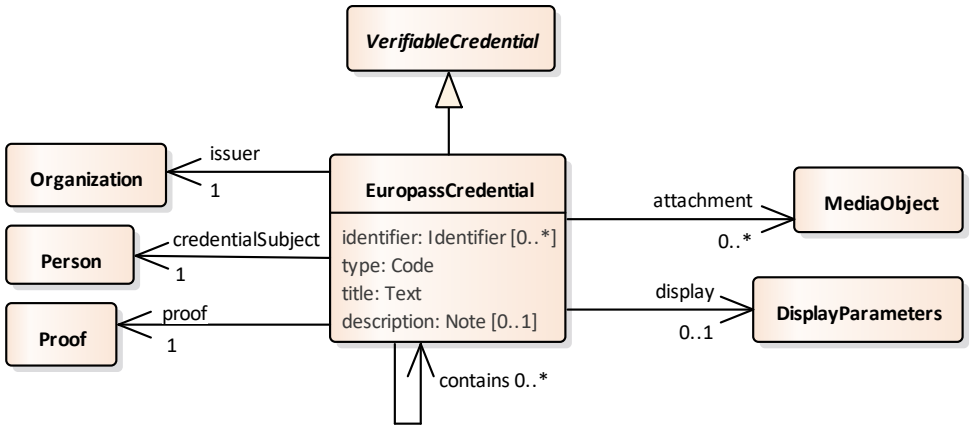


Table 21: EuropassCredential detailed class – definitions, data types and cardinalities

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Credential Identifier	An alternative identifier of the credential typically assigned to the credential by the issuing organisation.	identifier	Property Identifier	0 - *	
Type	The type of credential.	type	Property Code	1	Europass Standard List of Credential Types
Title	The full official title of the issued credential (maximum 50 characters).	title	Property Text	1	
Description	A summary of the claim or group of claims being made about a person (maximum 140 words).	description	Property Note	0. .1	
Issuer	The organisaton that issued the credential and sealed it with their digital e-seal.	issuer	Association Organisation	1	
Owner	The person about which claims are made	credential-Subject	Association Person	1	

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
	and who owns the credential				
Display Parameters	The display details of the credential.	display	Association DisplayParameters	0. .1	
Attachments	Any digital document (PDF, JPEG or PNG format) that an issuer has attached to the Europass document.	attachment	Association MediaObject	0 - *	
Proof	The cryptographic proofs that can be used to detect tampering and verify the authorship of a credential or presentation.	proof	Association Proof	1	
Sub-credentials	A credential embedded within the credential. Smaller sub-credentials (micro-credentials), that make up this larger credential when combined.	contains	Association EuropassCredential	0 - *	

Proof

Class description: The cryptographic proof that can be used to detect tampering and verify the authorship of a credential or presentation.

[This class is referenced from the class `EuropassCredential` via the property `proof`, but is not described with any properties on the Europass GitHub site.]

Agent < abstract >

Class description: An entity that is able to carry out actions.

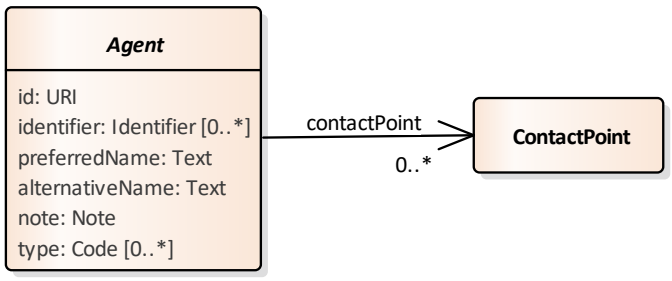


Table 22: Agent detailed class – definitions, data types and cardinalities

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Agent UID	A portable identifier of the agent.	id	ID/PK URI	1	
N/A	A formally-issued Identifier for the Agent.	identifier	Property Identifier	0 - *	
N/A	The Type of an Agent as described in a controlled vocabulary.	type	Property Code	0 - *	QMS List Of Organisation Types
Preferred Name	The primary name of the agent.	preferredName	Property Text	0. .1	
N/A	An agent may have any number of alternative or informal names.	alternativeName	Property Text	0 - *	
More information	An additional free text note about the agent.	note	Property Note	0 - *	
Contact information	The contact information of an agent.	contactPoint	Association ContactPoint	0 - *	

Organisation < extends Agent >

Class description: A legal person / registered organisation.

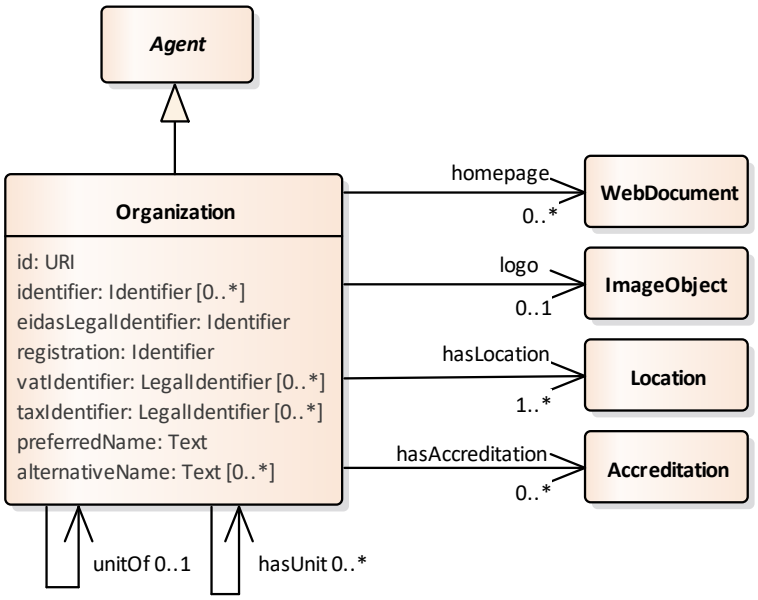


Table 23: Organisation detailed class – definitions, data types and cardinalities

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Organisation UID	The unique and portable identifier of the organisation	id	ID/PK URI	1	
Other Identifier	Another formally-issued identifier for the organisation.	identifier	Property Identifier	0 - *	
eIDAS Identifier	The official identification number of the organisation, as awarded by the relevant national authority.authority.[2]	eidasLegal-Identifier	Property Identifier	1	
Registration	The legal identifier of an organization. The identifier given to a registered organization by the authority with which it is registered. The legal status of a registered organization is conferred on it by an authority within a given jurisdiction. The Legal Identifier is therefore a fundamental relationship between an organization and the authority with which it is registered.	registration	Property Identifier	1	
VAT Number	The Value-Added Tax ID.	vatIdentifier	Property LegalIdentifier	0 - *	
Tax / Fiscal Identifier	Fiscal ID of the organisation	taxIdentifier	Property LegalIdentifier	0 - *	
Legal Name	The primary name of the organisation.	preferredName	Property Text	1	
Common Name	An (optional) alternative name of the organisation as typically used in	alternativeNam e	Property Text	0 - *	

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
	documents, including credentials.				
Homepage	A homepage about the organisation.	homepage	Association WebDocument	0 - *	
	The legally registered site of the organisation.	hasLocation	Association Location	1. .*	
Accreditation	Accreditation Records associated with the organisation. More information about the accreditation database is available here.	hasAccreditation	Association Accreditation	0 - *	
Child Organisation	A smaller organisation of which forms part of this organisation, e.g. a Department within a larger Organisation.	hasUnit	Association Organisation	0 - *	
Parent Organisation	Indicates a larger Organisation of which this Unit is a part of, e.g. the Organisation within which a Department operates.	unitOf	Association Organisation	0. .1	
Logo	The logo of the organisation	logo	Association ImageObject	0. .1	

[2]: See chapter 5.1.4 in Draft ETSI EN 319 412-1 V1.4.2

Person < extends Agent >

Class description: A human being.

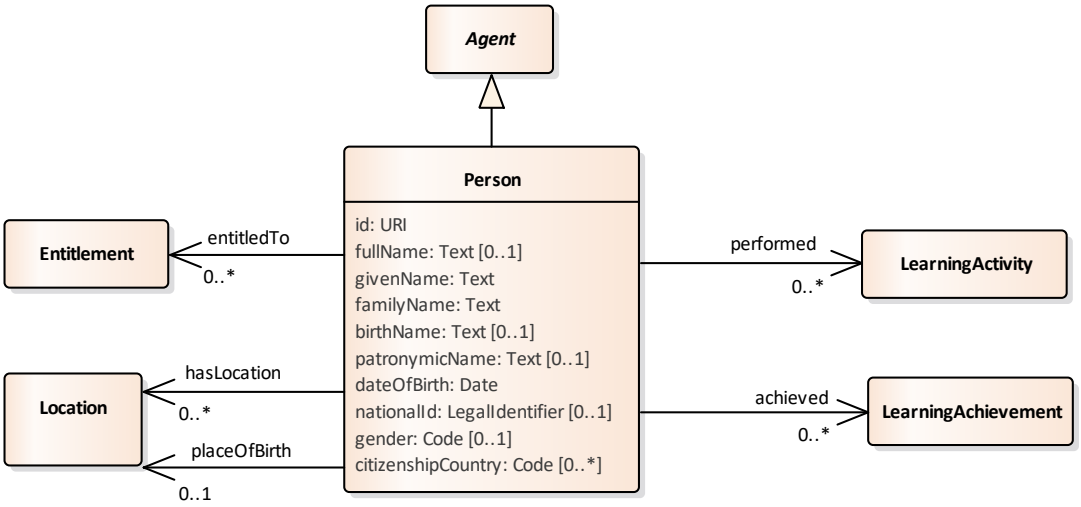


Table 24: Person detailed class – definitions, data types and cardinalities

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Person UID	The unique and portable identifier of the person	id	ID/PK URI	1	
National ID number	The "primary" national identifier of the person.	nationalId	Property LegalIdentifier	0. .1	
Other identifier(s)	An (optional) alternative formally-issued identifier for the person, e.g. social security number, student ID card number, to club membership, etc.	identifier	Property Identifier	0 - *	
Full name	The complete name of the person as one string.	fullName	Property (sub-Property of Agent prefferedName property) Text	0. .1	
Given name	The given name(s) of the person.	givenNames	Property Text	1	
Family name	The family name of the person.	familyName	Property Text	1	
Birth name	The name of the person at birth. Birth names tend to be persistent and for this reason they are	birthName	Property Text	0. .1	

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
	recorded by some public sector information systems. There is no granularity for birth name - the full name should be recorded in a single field.				
Patronymic name	Patronymic names are important in some countries. Iceland does not have a concept of 'family name' in the way that many other European countries do, for example, Erik Magnusson and Erika Magnusdottir are siblings, both offspring of Mangnus, irrespective of his patronymic name. In Bulgaria and Russia, patronymic names are in every day usage, for example, the Sergeyevich in 'Mikhail Sergeyevich Gorbachev.'	patronymic-Name	Property Text	0. .1	
Date of birth	The birth date of the person.	dateOfBirth	Property Date	1	
Place of birth	The place of birth of the person.	placeOfBirth	Property Location	0. .1	
Gender	The gender of the person.	gender	Property Code	0. .1	MDR Human Sex Named Authority List.
Citizenship	The country (or countries) that conferred	citizenship-Country	Property Code	0 - *	MDR Countries Named

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
	citizenship rights on the person				Authority List. NUTS.
Location	A location related to a Person. (e.g. a person's home or residence location, a person's work place location, site location of an organisation, etc.)	hasLocation	Association Location	0 - *	MDR Countries Named Authority List. NUTS (skossified and published version by ESCO).
Learning activities	A learning activity that a person participated in or attended	performed	Association LearningActivity	0 - *	
Learning achievements	An achievement of the person	achieved	Association LearningAchievement	0 - *	
Learning entitlements	The entitlement of the person	entitledTo	Association Entitlement	0 - *	

Learning Achievement

Class description: The acquisition of knowledge, skills or responsibility and autonomy. A recognised and/or awarded set of learning outcomes of an individual.

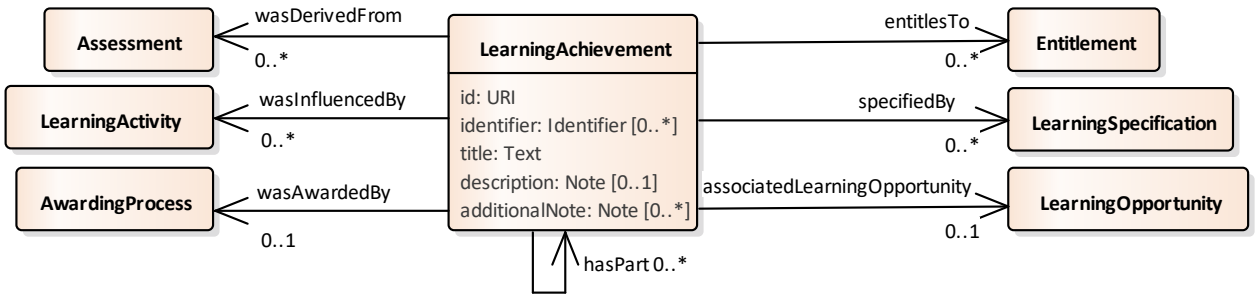


Table 25: LearningAchievement detailed class – definitions, data types and cardinalities

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Learning Achievement UID	A portable and identifier of the learning achievement	id	ID/PK URI	1	
Learning Achievement Identifier	An alternative identifier assigned to the achievement	identifier	Property Identifier	0 - *	

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
	by the organisation awarding the achievement.				
Title	The title of the achievement.	title	Property Text	1	
Description	A description of the achievement.	description	Property Note	0..1	
More information	An additional free text note about the achievement.	additionalNote	Property Note	0 - *	
Proven by	An assessment which proves the acquisition of the learning outcomes which make up the achievement.	wasDerivedFrom	Association Assessment	0 - *	
Influenced by	Activities which contributed to the acquisition of the learning outcomes which make up the achievement.	wasInfluencedBy	Association LearningActivity	0 - *	
Awarding Details	The awarding details of this achievement.	wasAwardedBy	Association AwardingProcess	0..1	
Sub-achievements	Smaller units of achievement, which when combined make up this achievement	hasPart	Association LearningAchievement	0 - *	
Entitles Owner to	Entitlements the owner has received as a result of this achievement	entitlesTo	Association Entitlement	0 - *	
N/A	What has been learned.	specifiedBy	Association LearningSpecification	0 - *	
Linked to Learning Opportunity	The learning opportunity that was taken to obtain the awarded LearningSpecification.	associatedLearningOpportunity	Association LearningOpportunity	0..1	

Learning Specification

Class description: A description of what a person may learn using the opportunity, expressed as learning outcomes. A specification of learning.

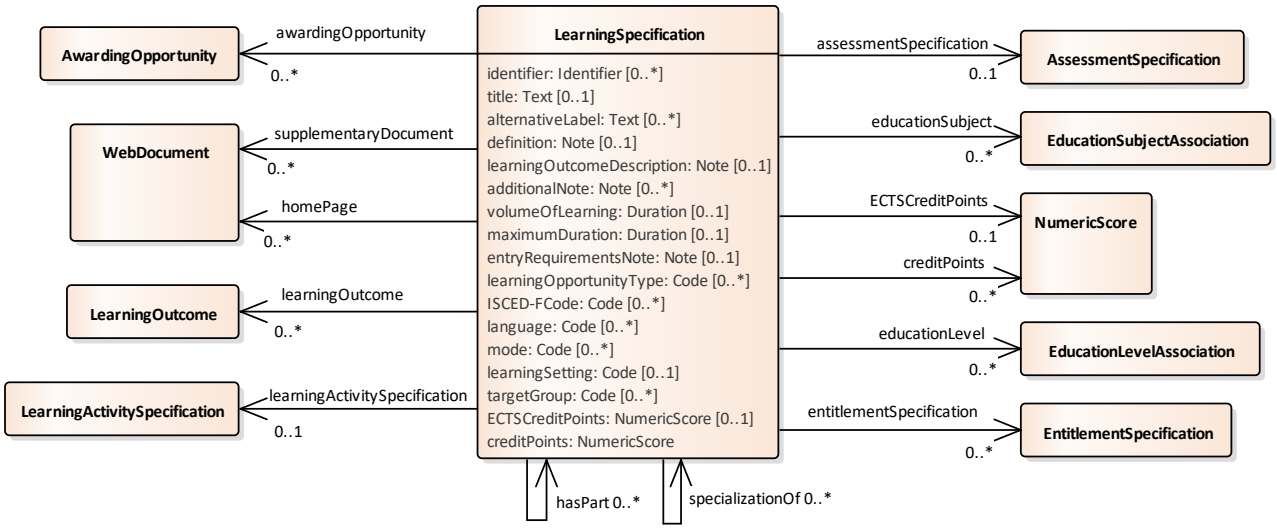


Table 26: LearningSpecification detailed class – definitions, data types and cardinalities

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Learning Specification UID	A portable and unique identifier of the learning specification.	id	ID/PK URI	1	
Learning specification Identifier	An alternative identifier of the learning specification, as assigned to it by the organisation who designed the specification.	identifier	Property Identifier	0 - *	
Learning Opportunity Type	The type of learning opportunity.	learning-OpportunityType	Property Code	0 - *	Europass Standard List of Learning Opportunity Types.
Title	The title of the learning specification	title	Property Text	0. .1	

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
N/A	An alternative name of the learning specification	alternativeLabel	Property Text	0 - *	
Description	Short and abstract description about the learning specification.	definition	Property Note	0. .1	
N/A	The full learning outcome description of the learning specification.	learningOutcome-Description	Property Note	0. .1	
More information	An additional free text note about the learning specification.	additionalNote	Property Note	0 - *	
Homepage	The homepage (a public web document) of the learning specification.	homePage	Association WebDocument	0 - *	
Other Documents	A public web document containing additional documentation about the learning specification.	supplementary-Document	Association WebDocument	0 - *	
Thematic Area	Thematic Area according to the ISCED-F 2013 Classification	ISCEDFCode	Property Code	0 - *	ISCED-F.
N/A	An associated field of education from another semantic framework than the ISCED classification.	educationSubject	Association Education-Subject-Association	0 - *	

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Volume of Learning	The estimated number of hours the learner is expected to spend engaged in learning to earn the award. This would include the notional number of hours in class, in group work, in practicals, as well as hours engaged in self-motivated study.	volumeOfLearning	Property Duration	0. .1	
ECTS Credit Points	The credit points assigned to the learning specification, following the ECTS credit system.	ECTSCreditPoints	Property NumericScore	0. .1	ECTS scoring scheme from Europass Standard List of Educational Credit Systems.
N/A	The credit points assigned to the learning specification, following an alternative educational credit system	creditPoints	Property NumericScore	0 - *	Europass Standard List of Educational Credit Systems
N/A	An associated level of education within a semantic framework describing education levels.	educationLevel	Association EducationLevel-Association	0 - *	
Language(s) of Instruction	The instruction and/or assessment language(s) used.	language	Property Code	0 - *	MDR Languages Named Authority List

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Mode of Learning	The mode of learning and or assessment	mode	Property Code	0 - *	Europass Standard List of Modes Of Learning and Assessment.
Learning Setting	The type of learning setting (formal, non-formal).	learningSetting	Property Code	0. .1	formal, non-formal.
Maximum Duration in Months	The maximum duration (in months) that a person may use to complete the learning opportunity.	maximumDuration	Property Duration	0. .1	
Target Group	A specific target group or category for which this specification is designed.	targetGroup	Property Code	0 - *	Europass Standard List of Target Groups.
Entry Requirements	Specific entry requirements or prerequisites of individuals for which this specification is designed to start this learning opportunity.	entry-Requirements-Note	Property Note	0. .1	
Learning Outcomes	An individual (expected) learning outcome of the learning specification.	learningOutcome	Association Learning-Outcome	0 - *	
Activities	Activities which a person can perform to acquire the expected learning outcomes	learningActivity-Specification	Association LearningActivity-Specification	0. .1	

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Assessments	Assessments a person can undergo to prove the acquisition of the learning outcomes	assessment-Specification	Association Assessment-Specification	0. .1	
Entitlements	Rights (such as which the person may acquire as a result of acquiring the learning outcomes)	entitlement-Specification	Association Entitlement-Specification	0 - *	
Awarding Information	Refers to an activity related to the awarding of the learning specification, such as the country or region where the qualifi-cation is awarded, the awarding body and optionally the awarding period now or in the past	awarding-Opportunity	Association Awarding-Opportunity	0 - *	
Learning Sub-Specifications	A learning specification can be composed of other "narrower" learning specifications which when combined make up this learning specification.	hasPart	Association Learning-Specification	0 - *	
N/A	A learning specification (e.g. a standard) of which this specification is a specialisation.[1]	specialisationOf	Association Learning-Specification	0 - *	

[1]: To be implemented at a later stage

Assessment

Class description: The result of a process establishing the extent to which a learner has attained particular knowledge, skills and competences against criteria such as learning outcomes or standards of competence.

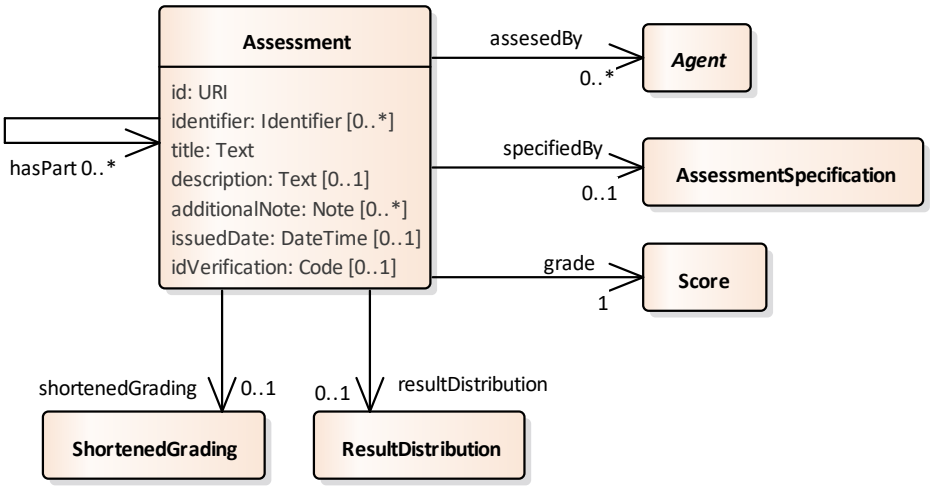


Table 27: Assessment detailed class – definitions, data types and cardinalities

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Assessment UID	A portable identifier of the assessment	id	ID/PK URI	1	
Assessment Identifier	An alternative identifier assigned to the assessment by the organisation grading the assessment	identifier	Property Identifier	0 - *	
Title	The title of the assessment.	title	Property Text	1	
Description	A description of the assessment.	description	Property Text	0. .1	
More information	An additional free text note about the assessment.	additionalNote	Property Note	0 - *	
Grade	A resulting grade of the assessment	grade	Property Score	1	
N/A	Indicator of how well the student was graded when compared to other students	shortened-Grading	Association Shortened-Grading	0. .1	

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
N/A	Describes a histogram of results achieved by all the students of a particular learning assessment.	result-Distribution	Association Result-Distribution	0. .1	
Assessment Date	Date the grade was awarded.	issuedDate	Property DateTime	0. .1	
Method of assessment, supervision and id verification	Method of assessment supervision and id verification.	idVerification	Property Code	0. .1	Europass Standard List of Methods Of Supervision And Verification.
Assessment conducted by	The competent body that awarded the grade	assessedBy	Association Agent	0 - *	
N/A	The specification of this assessment.	specifiedBy	Association Assessment-Specification	0. .1	
Sub-Assessments	Smaller assessments, which when combined make up and can influence this assessment	hasPart	Association Assessment	0 - *	

Assessment Specification

Class description: An Assessment Specification is a specification of a process establishing the extent to which a learner has attained particular knowledge, skills and competences against criteria such as learning outcomes or standards of competence.

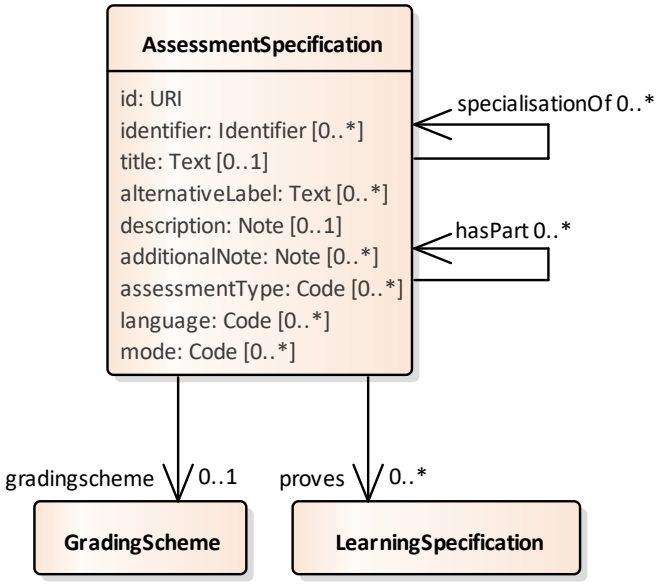


Table 28: Assessment Specification

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Assessment Specification UID	A portable and Unique Identifier of the Assessment Specification	id	ID/PK URI	1	
Assessment Specification Identifier	An alternative identifier of the assessment specification, as assigned to it by the organisation who designed the specification.	identifier	Property Identifier	0 - *	
Title	The title of the assessment specification.	title	Property Text	0. .1	
N/A	An alternative name of the assessment specification.	alternativeLabel	Property Text	0 - *	
Description	A free text description of the assessment specification.	description	Property Note	0. .1	
More information	An additional free text note about the	additionalNote	Property Note	0 - *	

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
	assessment specification.				
Homepage	The homepage (a public web document) describing the details of the assessment specification	homePage	Association WebDocument	0 - *	
Other Documents	A public web document containing additional documentation about the assessment specification.	supplementary-Document	Association WebDocument	0 - *	
Assessment Type	The type of assessment.	assessmentType	Property Code	0 - *	Europass Standard List of Assessment Types
Language of Assessment	The language(s) of assessment used.	language	Property Code	0 - *	MDR Languages Named Authority List
Mode of Assessment	The mode of learning and or assessment	mode	Property Code	0 - *	Europass Standard List of Modes Of Learning and Assessment
Grading Scheme	A description of the specification of which learning outcomes are or have been proven	gradingscheme	Association ScoringScheme *)	0. .1	
Demonstrates	The learning achievement (and related learning outcomes) this assessment is designed to test.	proves	Association Learning-Specification	0 - *	

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Assessment Sub-Specifications	A assessment specification can be composed of other "narrower" assessment specifications which when combined make up this assessment specification.	hasPart	Association Assessment-Specification	0 - *	
N/A	An assessment specification (e.g. a standard) of which this specification is a specialisation.	specialisationOf	Association Assessment-Specification	0 - *	

*) There is no class named 'ScoringScheme'. It is assumed that the range given should have been 'GradingScheme'.]

Grading Scheme

Class description: A set of criteria that measures varying levels of achievement.

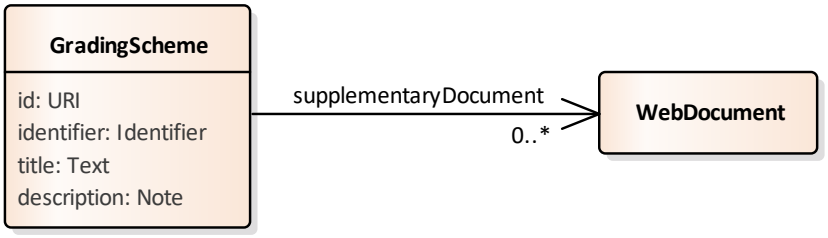


Table 29: GradingScheme detailed class – definitions, data types and cardinalities

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Grading Scheme UID	A portable and unique identifier of the Grading Scheme.	id	ID/PK URI	1	
Grading Scheme Identifier	An alternative identifier of the Grading Scheme assigned to it by the organisation administering the scheme.	identifier	Property Identifier	0 - *	
Title	The title of the scoring scheme.	title	Property Text	0. .1	

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Description	A free text describing the scoring scheme.	description	Property Note	0. .1	
Other Documents	A public web document containing additional documentation about the scoring system.	supplementary-Document	Association WebDocument	0 - *	

Awarding Process

Class description: The process of an organisation awarding Learning Achievement to person based on a Learning Specification (e.g. a qualification). It is used to specify the organisation that awarded the LearningSpecification to the individual, the country or region where the LearningSpecification was awarded, and optionally the date of awarding.

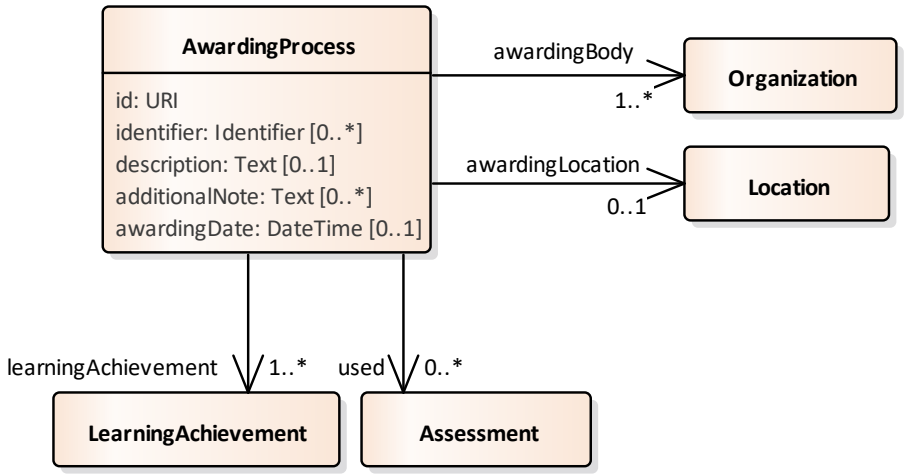


Table 30: AwardingProcess detailed class – definitions, data types and cardinalities

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Awarding Process UID	A portable and Unique Identifier of the Awarding Process.	id	ID/PK URI	1	
N/A	An alternative identifier of the awarding process.	identifier	Property Identifier	0 - *	
Description	A description of the awarding process to the individual.	description	Property Text	0. .1	

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
More information	An additional free text note (e.g. a comment, a remark, etc.)	additional-Note	Property Text	0 - *	
Assessment utilised	The assessment that provided the basis for this awarding.	used	Association Assessment	0 - *	
Learning achievement	The resulting learning achievement	learning-Achievement	Association Learning-Achievement	1. .*	
Awarding organisation	The awarding body that awarded the Achievement to the individual. Only in cases of co-awarding/co-graduation, where a qualification award is issued to an individual by two or more organisations the cardinality is greater than 1.	awardingBody	Association Organisation	1. .*	
Location	The location where the awarding activity took place (country/region where the qualification was awarded).	awarding-Location	Association Location	0. .1	MDR Countries Named Authority List. NUTS (skossified and published version by ESCO).
Awarding Date	The date when the LearningSpecification was awarded. If not specified it is undefined (“not known”).	awardingDate	Property DateTime	0. .1	

Location

Class description: An identifiable geographic place.

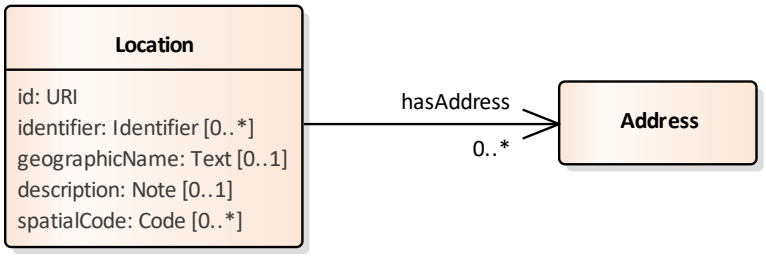


Table 31: Location detailed class – definitions, data types and cardinalities

Label	Definition	Field	Type Range (data type)	Card	Recommended RSA
Location UID	A portable identifier of the location.	id	ID/PK URI	1	
	A location identifier.	Identifier *)	Property Identifier	0 - *	
Name	A proper noun applied to a spatial object.	geographicName	Property Text	0. .1	
Location	A code identifying a spatial scope in which this physical location is located.	spatialCode	Property Code	0 - *	MDR Countries Named Authority List. NUTS. MDR Place Named Authority List.
Description	A free text describing the location.	description	Property Note	0. .1	
Address	An address associated with the location.	hasAddress	Association Address	0 - *	

*) The field name was missing in the EDCI documentation at the time the class decription was copyed. It is assumed that the field name is: 'identifier'.

Address

Class description: An address.

Web Document

Class description: A public web document.

Shortened Grading

Class description: Indicator of *how well* the student was graded when compared to other students.

Result Distribution

Class description: Describes a histogram of results achieved by all the students of this course instance.

Learning Outcome

Class description: a statement regarding what a learner knows, understands and is able to do on completion of a learning process, which are defined in terms of knowledge, skills and responsibility and autonomy.

Awarding Opportunity

Class description: An awarding activity represents an activity related to the awarding of a LearningSpecification. It is used to specify the country or region where the LearningSpecification is awarded, the awarding body and optionally the awarding period now or in the past.

Learning Activity Specification

Class description: The specification of a process which leads to the acquisition of knowledge, skills or responsibility and autonomy.

Education Subject Association

A class by this name is used as range for the property 'educationSubject' on the class LearningSpecification. Such a class was not found in the documentation. It might be that it was the intention that it should have been 'AssociationObject' that is described below.

Association Object

Class description: The details of an association or an alignment between a resource and another node in an established semantic framework. This class can be used to relate, annotate or align a resource to another semantic asset. Described in the QMS.

Education Level Association

A class by this name is used as range for the property 'educationLevel' on the class LearningSpecification. Such a class was not found in the documentation.

Entitlement Specification

Class description: The specification of a right a person has access to, typically as a result of a learning achievement. It may take the form of the right to be a member of an organisation, to follow a certain learning opportunity specification, or to follow a certain career.

Learning Activity

Class description: Any process which leads to the acquisition of knowledge, skills or responsibility and autonomy.

Learning Opportunity

Class description: An opportunity to realise a given set of learning outcomes via a learning activity and/or assessment.

Entitlement

Class description: A right, e.g. to practice a profession, take advantage of a learning opportunity or join an organisation, as a result of the acquisition of knowledge, skills, responsibility and/or autonomy.

Contact point

Class description: Details to Contact an Agent. A contact point for an agent.

Annex I-c. EDCI defined structured data types

Identifier - Composite Type

Definition: A character string used to identify a resource.

An identifier is a character string used to uniquely identify one instance of an object within an identification scheme that is managed by an agency.

The Identifier class is a critical aspect of the edci model. It is used to identify persons and organisations. In these cases and more, the identifier itself will be some sort of alpha-numeric string but that string only has meaning if it is contextualised.

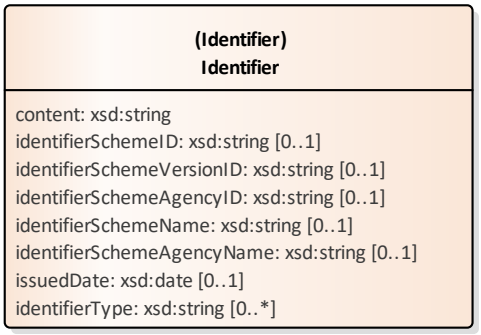


Table 32: Identifier type detailed class – definitions, data types and cardinalities

Data Type	Field	Type Range (data type)	Card	Definition	Desciption
Identifier	Content	Attribute String	1	Content string which is the identifier.	Content string which is the identifier. A character string used to uniquely identify one instance of an object within an identification scheme that is managed by an agency.
Identifier	Identifier Scheme ID	Attribute String	0. .1	Identification of the identifier scheme.	Identification of the identifier scheme. The identifier register (the managing/originating system of the identifier). This can be seen as the namespace in which the assigned identifier is unique.
Identifier	Identifier Scheme Version ID	Attribute String	0. .1	Identification of the version of the identifier scheme	Identification of the version of the identifier scheme.
Identifier	Identifier Scheme Agency ID	Attribute String	0. .1	Identification of the agent that manages the identifier scheme.	Identification of the agent that manages the identifier scheme. The agent that issued the identifier. (e.g. a URI)

Data Type	Field	Type Range (data type)	Card	Definition	Desciption
Identifier	Identifier Scheme Name	Attribute String	0. .1	The name of the identifier scheme.	The name of the identifier scheme.
Identifier	Identifier Scheme Agency Name	Attribute String	0. .1	The name of the agent that manages the identifier scheme	The name of the agent that manages the identifier scheme. The agent that issued the identifier.
Identifier	Issued Date	Attribute Date	0. .1	The date on which the identifier was issued	The date on which the identifier was issued
Identifier	Identifier Type	Attribute String	*	A code used to classify the type of identifier	A code used to classify the type of identifier

LegalIdentifier - Composite Type < extends Identifier >

Definition: A legal identifier. A legal identifier is a formally issued identifier by a given authorithy within a given jurisdiction. The identifier has a spatial context.

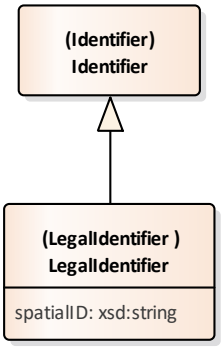


Table 33: LegalIdentifier type detailed class – definitions, data types and cardinalities

Data Type	Field	Type Range (data type)	Card	Definition	Desciption	Recommended RSA
Identifier	spatial ID	Attribute String	1	The identifier of the country and/or jurisdiction.	The identifier of the country and/or jurisdiction.	MDR Countries Named Authority Lis. NUTS

Code - Composite Type

Definition: A term from a controlled vocabulary. (a code from a code list) Interoperability between data sets is incr0065ased dramatically when terms from controlled vocabularies are used in favour of free text. The conceptual Code data type is used wherever one or more controlled vocabularies are known to exist for a particular domain of interest. It is not the job of the JV/CV Vocabularies to mandate which controlled vocabularies are used but guidance on how to use them is provided.

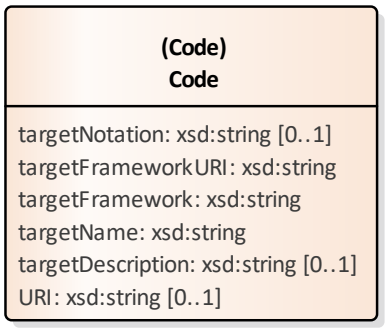


Table 34: Code type detailed class – definitions, data types and cardinalities

Data Type	Field	Type Range (data type)	Card	Definition	Desciption
Code	Target Notation	Attribute String	0. .1		The term.
Code	Target Framework URI	Attribute String	1	The identification of the controlled vocabulary.	The identification of the controlled vocabulary (the code list). (e.g. a URI)
Code	Target Framework	Text String	1	The name of the controlled vocabulary	The name of the controlled vocabulary (the code list).
Code	Target Name	Text String	1	The text equivalent of the code content component.	The text equivalent of the code content component.
Code	Target Description	Text String	0. .1		
Code	URI	Attribute String	0. .1	A portable identifier (i.e a URI) of the code.	A portable identifier (i.e a URI) of the code.

Text - Composite Type

Definition: A character string (i.e. a finite set of characters) generally in the form of words of a language.

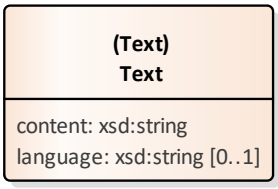


Table 35: Text type detailed class – definitions, data types and cardinalities

Data Type	Field	Type Range (data type)	Card	Definition	Desciption
Text	Content	Attribute String	1	The character string.	The character string.
Text	Language	Attribute String	0. .1	The identifier of the language used in the Content attribute	The identifier of the language used in the Content attribute

Note - Composite Type

Definition: A formatted character string (i.e. a finite set of characters) generally in the form of words of a language. The character string is passed/included in, and can be represented as, a (formatted) document fragment (formatted) according a given mimetype (e.g. "text/plain", "text/html", etc.)

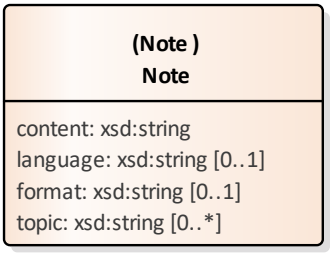


Table 36: Note type detailed class – definitions, data types and cardinalities

Data Type	Field	Type Range (data type)	Card	Definition	Desciption
Note	Content	Attribute String	1	The free text note.	The free text note
Note	Language	Attribute String	0. .1	The identifier of the language used in the Content attribute.	The identifier of the language used in the Content attribute.
Note	Format	Attribute String	0. .1	The identifier of the mimetype used in the Content attribute.	The identifier of the mimetype used in the Content attribute.
Note	Topic	Attribute String	0 - *	The information topic this note is about.	The information topic this note is about.

Notation - Composite Type

Definition: A notation (or code) is a character string according a given syntax encoding scheme.

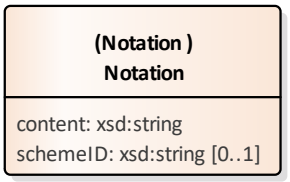


Table 37: Notation type detailed class – definitions, data types and cardinalities

Data Type	Field	Type Range (data type)	Card	Definition	Desciption
Notation	Content	Attribute String	1	A notation (or code).	A notation (or code).
Notation	Scheme ID	Attribute String	0. .1	The syntax encoding scheme.	The syntax encoding scheme. A particular system of notations or classification codes.

Score - Composite Type

Definition: A score.

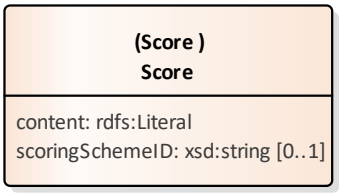


Table 38: Score type detailed class – definitions, data types and cardinalities

Data Type	Field	Type Range (data type)	Card	Definition	Desciption
Score	Content	Attribute Literal	1	The score	The score.
Score	Scoring Scheme ID	Attribute String	0. .1	The identifier of the scoring scheme used in the Content attribute.	The identifier of the scoring scheme used in the Content attribute. Refers to the type of scoring methodology or convention.

NumericScore < extends Score >

Definition: A numeric score.

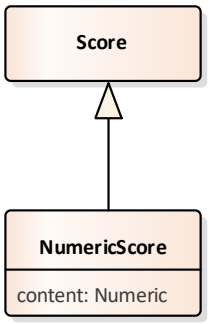


Table 39: NumericScore type detailed class – definitions, data types and cardinalities

Data Type	Field	Type Range (data type)	Card	Definition	Desciption	Recommended RSA
NumericScore	Content	Attribute Numeric	1	The numeric score	The numeric score	

TextScore < extends Score >

Definition: A textual score.

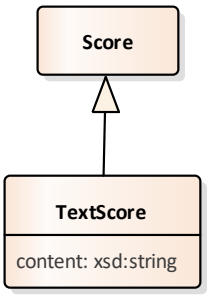


Table 40: TextScore type detailed class – definitions, data types and cardinalities

Data Type	Field	Type Range (data type)	Card	Definition	Desciption
TextScore	Content	Attribute String	1	The textual score.	The textual score.

Amount - Composite Type

Definition: An amount.

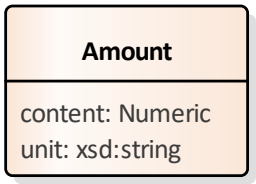


Table 41: Amount type detailed class – definitions, data types and cardinalities

Data Type	Field	Type Range (data type)	Card	Definition	Desciption	Recommended RSA
Amount	Content	Attribute Numeric	1	The numeric value (i.e. price, salary, etc.).		
Amount	Unit	Attribute String	1	A code indicating the currency in which the amount is indicated/expressed		MDR Currencies Named Authority List

EDCI used primitive types

Table 42: EDCI used primitive types detailed class – definitions, data types and cardinalities

Data Type	Range (data type)	Definition
String	xsd:string	A character string, i.e., a finite set of characters.
DateTime	xsd:dateTime	A date designating a point in time. Specifies a date and a time.
Date	xsd:date	A date
URI	xsd:anyURI	A Uniform Resource Identifier
IndicatorType	xsd:boolean	A boolean indicating true or false.
PercentType	xsd:decimal	A rate, number or proportion per hundred.
PositiveInteger	xsd:positiveInteger	A positive integer.
Numeric	xsd:decimal	A numeric value.
Duration	xsd:duration	*)

**) Notice that while EDCI uses xsd:duration for RDF use it would be recommendable to use either xdt:yearMonthDuration or xdt:dayTimeDuration [34].*

[34] https://www.w3.org/TR/2006/NOTE-swbp-xsch-datatypes-20060314/#section-duration

Annex II. IAL Specification API

```yaml
openapi: 3.0.3
info:
  description: DE4A IDK Swagger for the IAL API
  version: 1.0.0
  title: Swagger DEA4 IDK Issuing Authority Locator
  contact:
    email: apiteam@de4a.eu
  license:
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
servers:
  - url: https://idk.de4a.eu/v1
    description: fake production server

paths:
  /ial/{canonicalEvidenceTypeId}:
    get:
      parameters:
        - name: canonicalEvidenceTypeId
          in: path
          required: true
          description: evidence type with a canonical definition
          schema:
            $ref: '#/components/schemas/CanonicalEvidenceType'
          example: BirthCertificate
      responses:
        '200':
          description: OK
          content:
            application/json charset=utf-8:
              schema:
                anyOf:
                  - $ref: '#/components/schemas/AvailableSources'
        '5XX':
          description: Unexpected error
        '400':
          description: Bad request. Evidence Type ID must be
              - HigherEdCertificate
              - SecondaryEdCertificate
              - ResidencyProof
              - BirthCertificate
              - MarriageCertificate
              - CompanyRegistration

  /ial/{canonicalEvidenceTypeId}/{countryCode}:
    get:
      parameters:
        - name: canonicalEvidenceTypeId
          in: path
          required: true
          description: evidence type with a canonical definition
          schema:
            $ref: '#/components/schemas/CanonicalEvidenceType'
          example: BirthCertificate
        - name: countryCode
          in: path
```

```
        required: true
        description: country of the available sources
        schema:
          type: string
          pattern: '[A-Z][A-Z]'
        example: ES
    responses:
      '200':
        description: OK
        content:
          application/json charset=utf-8:
            schema:
              type: object
              properties:
                atuLevel:
                  $ref: '#/components/schemas/AtuLevel'
                provision:
                    $ref: '#/components/schemas/ProvisionItem'
      '5XX':
        description: Unexpected error
      '400':
        description: Bad request. Evidence Type ID must be
            - HigherEdCertificate
            - SecondaryEdCertificate
            - ResidencyProof
            - BirthCertificate
            - MarriageCertificate
            - CompanyRegistration
      '404':
        description: Not found country code

/provision:
  get:
    parameters:
      - name: canonicalEvidenceTypeId
        in: query
        required: true
        description: evidence type with a canonical definition
        schema:
          $ref: '#/components/schemas/CanonicalEvidenceType'
        example: BirthCertificate
      - name: dataOwnerId
        in: query
        required: true
        description: country of the available sources
        schema:
          type: string
          pattern: 'iso6523-actorid-upis::[0-9][0-9][0-9][0-9]:[A-Z0-9]{1,15}'
        example: iso6523-actorid-upis::9991:LU000000025
    responses:
      '200':
        description: OK
        content:
          application/json charset=utf-8:
            schema:
              type: object
              properties:
                atuLevel:
```

```
                  $ref: '#/components/schemas/AtuLevel'
                provision:                      $ref:
'#/components/schemas/Provision'
        '5XX':
          description: Unexpected error
        '400':
          description: Bad request. Evidence Type ID must be
              - HigherEdCertificate
              - SecondaryEdCertificate
              - ResidencyProof
              - BirthCertificate
              - MarriageCertificate
              - CompanyRegistration
        '404':
          description: Not found country code

components:
  schemas:

    AvailableSources:
      type: array
      description: list of available sources for a canonical evidence type
organised by country
      example:
          - countryCode: ES
            atuLevel: lau
            provisions:
              numProvisions: 8123
              organisation: ES/nuts2/nuts3
          - countryCode: SI
            atuLevel: nuts0
            provisions:
              atuCode: SI031
              atuLatinName: Mura
              dataOwnerId: iso6523-actorid-upis::9991:SI990000105
              dataOwnerPrefLabel: Vlada Mure
              provisionType: usip
              redirectURL : https://moai.gov.si/usip
          - countryCode: LU
            atuLevel: nuts0
            provisions:
              dataOwnerId: iso6523-actorid-upis::9991:LU000000025
              dataOwnerPrefLabel: CENTRE DES TECHNOLOGIES DE L'INFORMATION DE
L'ETAT
              atuCode: LU
              atuLatinName: LUXEMBOURG
              provisionType: ip
          - countryCode: SI
            atuLevel: nuts0
            provisions:
              dataOwnerId: iso6523-actorid-upis::9991:SI990000105
              params:
                title: SI/nuts3
                paramset:
                  - SI/SI031
                  - SI/SI034
              dataOwnerPrefLabel: Minister za notranje zadeve
              atuCode: SI
```

```
              atuLatinName: SLOVENIJA
              provisionType: ip
      items:
        type: object
        properties:
          countryCode:
            type: string
            description: iso 3166 Alpha-2
            example: ES
          atuLevel:
            $ref: '#/components/schemas/AtuLevel'
          provisions:
            oneOf:
            - type: object
              description: with too many provisions
              required:
                - numProvisions
                - atuLevel
              properties:
                numProvisions:
                  type: integer
                  nullable: false
                  example: 8123
                atuLevel:
                  $ref: '#/components/schemas/AtuLevel'
                organisation:
                  type: string
                  description: upper-level territorial levels that help to select
the ATU provision
                  nullable: false
                  example: ES/nuts2/nuts3
            - type: array
              items:
                $ref: '#/components/schemas/ProvisionItem'

    ProvisionItem:
      type: object
      properties:
        atuCode:
          type: string
          example: ES
        atuLatinName:
          type: string
          example: España
        dataOwnerId:
          type: string
          example: urn:de4a-eu:provision::9920:ESS2833002E:BirthCertificate
        dataOwnerPrefLabel:
          type: string
          example: Ministerio de Justicia
        provision:
          $ref: '#/components/schemas/Provision'
    Provision:
          oneOf:
            - type: object
              description: user supported intermediation pattern provision
              required:
                - provisionType
```

```
            - redirectURL
          properties:
            provisionType:
              type: string
              nullable: false
              enum:
                - usip
              example: usip
            redirectURL:
              type: string
              nullable: false
              example: https://ctie.lu/usip
        - type: object
          description: intermediation pattern provision
          required:
            - provisionType
          properties:
            provisionType:
              type: string
              nullable: false
              enum:
                - ip
              example: ip
            params:
              type: array
              nullable: false
              items:
                type: object
                required:
                  - title
                properties:
                  title:
                    type: string
                    nullable: false
                    example: SI/nuts3
                  paramset:
                    type: array
                    nullable: false
                    items:
                      type: string
                      example: SI031
      discriminator:
        propertyName: provisionType

  AtuLevel:
    type: string
    enum:
      - nuts0
      - nuts1
      - nuts2
      - nuts3
      - lau
      - edu
    example: nuts0

  CanonicalEvidenceType:
    type: string
    enum:
```

- HigherEdCertificate
- SecondaryEdCertificate
- ResidencyProof
- BirthCertificate
- MarriageCertificate
- CompanyRegistration