AI4EU Deliverable D3.4

Manual on platform knowledge modelling

WP	3	Management and enrichment of the European AI on demand platform
Task	3.5	Providing and modelling knowledge on AI

Dissemination level[1]	PU
Nature[2]	R

Due delivery date	31/12/2020
Actual delivery date	

Lead beneficiary	DEM
Contributing beneficiaries	FhG, INRIA

Document Version	**Date**	**Author**	**Comments**[3]
v1.1	05/11/2019	Antonis Koukourikos, Costas Spyropoulos, Charis Akasiadis (DEM)	Analysis of Review comments
v1.2	19/11/2019	Antonis Koukourikos, Costas Spyropoulos (DEM)	Revised ToC / Section 2 & 3 updates
v1.3	29/11/2019	Charis Akasiadis (DEM)	Section 2 updates
v1.9	15/12/2019	Antonis Koukourikos, Costas Spyropoulos (DEM)	Document finalization
v2.0	18/12/2019	Antonis Koukourikos (DEM)	Appendix

[1] Dissemination level: **PU** = Public, **PP** = Restricted to other programme participants (including the JU), **RE** = Restricted to a group specified by the consortium (including the JU), **CO** = Confidential, only for members of the consortium (including the JU)
[2] Nature of the deliverable: **R** = Report, **P** = Prototype, **D** = Demonstrator, **O** = Other
[3] Creation, modification, final version for evaluation, revised version following evaluation, final

Deliverable Abstract

The AI4EU platform aims to constitute a one-stop-shop for European AI. End-users of any type will have the opportunity to discover and learn about various AI resources, which can be then integrated into a variety of everyday processes. To achieve this, a conceptual framework equipped with semantic technologies should be utilized. This deliverable presents the work performed in the scope of Task 3.5. It presents our ideas on modelling knowledge on AI and describes the state of the art regarding semantic web technologies, specifications, as well as semantic models and ontologies used in other relevant H2020 projects. The AI4EU Conceptual Semantic Model is designed in the form of an ontology, which, apart from introducing new concepts related to AI resources, also relies on widely used ontologies. Based on the AI4EU conceptual model and the corresponding ontology, a knowledge graph is designed and deployed, using the RDF and RDFS W3C specifications. The proposed model is evaluated in terms of consistency and expressivity, and the AI4EU Data model is compiled to serve as the basis for interconnecting different AI related platforms with AI4EU. Finally, a list of potential risks and barriers is provided, for designers to take into account for future improvements and extensions.

Deliverable Review

	Reviewer #1:			Reviewer #2:		
	Answer	**Comments**	**Type***	**Answer**	**Comments**	**Type***
1. Is the deliverable in accordance with						
(i) the Description of the Action?	☐ Yes ☐ No		☐ M ☐ m ☐ a	☐ Yes ☐ No		☐ M ☐ m ☐ a
(ii) the international State of the Art?	☐ Yes ☐ No		☐ M ☐ m ☐ a	☐ Yes ☐ No		☐ M ☐ m ☐ a
2. Is the quality of the deliverable in a status						
(i) that allows it to be sent to European Commission?	☐ Yes ☐ No		☐ M ☐ m ☐ a	☐ Yes ☐ No		☐ M ☐ m ☐ a
(ii) that needs improvement of the writing by the originator of the deliverable?	☐ Yes ☐ No		☐ M ☐ m ☐ a	☐ Yes ☐ No		☐ M ☐ m ☐ a
(iii) that needs further work by the Partners responsible for the deliverable?	☐ Yes ☐ No		☐ M ☐ m ☐ a	☐ Yes ☐ No		☐ M ☐ m ☐ a

* Type of comments: M = Major comment; m = minor comment; a = advice

1 Introduction

As part of Work Package 3, Task 3.5 aims to establish the conceptual foundation of the AI4EU platform. Its detailed description in the AI4EU Description of Action:

> **Task 3.5: Providing and modelling knowledge on AI**
>
> In this task, the knowledge for the collected AI resources will be modelled, stored and exposed using a semantic approach. One important activity is the creation of graph-based data model (e.g., an ontology and knowledge graph) to capitalise on the search engine functionalities (see T2.3) and to respond appropriately to users' queries. Based on previously established work on the conceptualisation of software, data, machine learning processes etc., the AI4EU ontology, a specification using Semantic Web technologies (like the RDF / RDF-S and the Web Ontology Language specifications), will be designed and created. The ontology will be improved and enriched through several iterations during the project. The process will entail inconsistency and non-satisfiability checking to ensure that the representation is semantically complete and valid.

The rationale for including the aforementioned activities relies on the observation that a conceptual framework, which allows the formalised description and association of AI resources, is essential for ensuring the stability, cohesiveness and expandability of the platform, also with respect to interoperability with other major AI-related EU H2020 platforms such as ELG, BEAT, etc.

In this context, we consider as a resource any entity that can be used for obtaining knowledge or technology around AI, as detailed in the conceptual model analysis in Section 2. The main goal of the relevant D3.4 deliverable is thus to build and document the data model underlying the collection of AI resources to be referenceable from within the platform. Given the foundational role of the data model in the design of the AI4EU knowledge base and the services built upon it, work on the model and the ontology started and produced results early on. The conceptual model was the foundation of the AI4EU knowledge graph, i.e., a repository that allows injection of descriptions and building search and visualization mechanisms on top of it. Most importantly, the semantic schema serves as the conceptual base of the AI4EU data model that supports the current implementation of the AI4EU platform, ensuring the coherence and cohesiveness of information in the platform and allowing semantically valid connections with external platforms. The present report therefore aims to present the design principles followed for the development of the ontology, its usage in the AI4EU knowledge graph, and the subsequent development of the ontology-based core data model that is used for building AI4EU platform services and integrating the platform with other resource description initiatives.

The document is structured as follows: Section 2 summarises the rationale behind the decision to adopt Semantic Web technologies. Section 3 provides an overview of the AI4EU conceptual model and discusses on its representation as an ontology using Semantic Web specifications. Following that, section 4 presents the design of the AI4EU knowledge graph and the functionalities it provides. Section 5 reports on the evaluation processes put in place to ensure the consistency and integrity of the AI4EU ontology and knowledge base. Section 6 provides details on the core data model that is based on the AI4EU ontology and constitutes the foundation for interoperating with other AI4EU work packages (namely WP2), as well as other initiatives that specialize in domain-specific AI resources, like the alignment with ELG presented in Section 7. Following the presentation of the complete picture of T3.5 activities, section 8 provides an overview of the expected risks stemming from the adopted approach. Section 9 concludes the report, providing a summary of all major points discussed.

2 Motivation and Related Work

Given the requirements for the AI4EU project, the usage of Semantic Web technologies was deemed as the best available choice for expressing the model and, consequently, the descriptions of the resources under the model.

Specifications and standards belonging to the Semantic Web family bring several advantages to our case:

- They provide a robust descriptive and axiomatic framework for defining concepts and their relations.
- Several editing and publishing tools are available.
- They are supported by a multitude of databases.
- Data expressed in them are readily consumable by web services and applications.
- Significant work for various domains relevant to AI4EU has already been carried out, hence there are reusable concept definitions that can be exploited for building the AI4EU data model.

The central specification that serves the majority of Semantic Web technologies is RDF[4], the W3C Resource Description Framework. The central notion of RDF is the triple, i.e., a Subject-Object-Predicate construct that declares a relation (Predicate) between two entities (the Subject and Object). Such relations can be defined at the conceptual level, between types of resources. This is where an ontology comes at hand, as it is able to define types (Classes) of resources as well as relations (Properties) between them, while also defining additional characteristics for the classes and properties via axioms, i.e., inference rules that must be held in order for a specification to be consistent. It follows that RDF and adjacent technologies constitute a well-established, heavily supported and web-ready framework for developing the AI4EU conceptual model and the respective knowledge base of resources.

Apart from relevant specifications, focus was also put on existing European AI-related platforms and their capabilities. In particular, we examined the work performed in the scope of three different H2020 projects, BEAT, BONSEYES, and ELG.

2.1 BEAT

2.1.1 Brief platform description

The BEAT platform[5] is an open-source solution that focuses on the sharing of AI-based experimental methods and results. It supports privacy by design and offers configurable confidentiality, as well as the reuse of hybrid algorithms. Its aim is to provide an alternative paradigm for the development and evaluation of methods involving machine learning and pattern recognition tools. The platform can be used for benchmarking such tools and components, and to perform objective and reproducible comparisons and evaluations.

[4] https://www.w3.org/RDF/

[5] https://www.beat-eu.org

BEAT offers an easy-to-use web-based experiment configurator where users can compose the experiment by combining available algorithms and databases that fit together with respect to configuration, input and output data format etc. in a visual manner [1].

2.1.2 Data model description

The main type of resources hosted by BEAT is *Experiments*. Experiments are defined in the form of toolchains, which comprise *Datasets*, *Blocks* and *Analysers*. Datasets are actually points of access to databases that contain the actual data used as input by Blocks. The latter represent the algorithm that the user selects for processing the data in an experiment. Finally, Analysers are other algorithms that evaluate the overall performance and generate the results. Note that toolchains can be used in more than one experiment.

BEAT uses an additional model for describing data formats. For databases and algorithms to correctly work together, the user must define the format of the exchange data in a form of a JSON object, with all the corresponding fields as keys and the primitive data types that are available, i.e., integers, unsigned integers, Boolean values, string, etc. A user may also reuse an existing data format used by some database or block.

BEAT uses these data models to verify that a toolchain is valid and to allow users to search for experiments. Result filtering is also available based on database name, protocol used, algorithms, analysers, experiment name, authors, dates, results, as well as toolchains and their authors.

2.2 BONSEYES

2.2.1 Brief platform description

The BONSEYES project aims to transform the development model for AI applications to an edge device-centric model, by offering an open and expandable platform, consisting of a data marketplace, a deep learning toolbox and universal developer reference platforms [2]. The data marketplace will enhance the AI systems development value chain since it will make data available to a larger audience comprising developers, data scientists, etc. By using the deep learning toolbox users will be able to select off-the-shelf deep learning methods to deploy for their applications, which can also be executed on edge, resource-restricted platforms, such as embedded systems, i.e., the developer reference platforms.

2.2.2 Data model description

Although there is not yet an official description of the underlying data model, the work of [6] focuses on AI compatibility and refers to three key concepts that are of interest for data scientists in AI, i.e., challenges--- problems or application areas, artifacts---data or models used for solving the challenge, and licenses. These in turn, are characterized by their type of compatibility, where the vertical type combines challenges and artifacts, as well as artifacts and licenses, and the horizontal compatibility type addresses compatibility between artifacts, and between licenses. In [6], a high-level process for artifact registration is also described, that is the definition and the upload of resources by the end-user.

Moreover, for BONSEYES, the context of an AI pipeline (i.e., a set of interconnected artifacts that fulfil challenges) is further linked to users, resources, and tools, which are, respectively, the user accounts that have access to the pipeline, the physical resources required for executing the pipeline, and the various software components utilized, which make effective use of the artifacts.

2.3 European Language Grid

2.3.1 Brief platform description

The European Language Grid (ELG)[6] is a European project that aims to create a European service platform for language technologies. This sector is the domain of interest for thousands of SMEs but is still quite fragmented due to the many different nations and the various languages spoken across Europe. ELG offers a scalable cloud-based platform that includes a large number of commercial and non-commercial tools, services, datasets, and resources, for all European languages, to help facilitate wide adoption of language technologies. This platform will allow users to upload and deploy language technology-related tools and datasets, and to also connect them with other resources that are present in the grid.

2.3.2 Data model description

ELG defines the "ELG-SHARE" Metadata Schema [3], which is linked to the META-SHARE and OMTD-SHARE ontologies. This schema describes the variety of resources found in the ELG platform, i.e., functional, such as tools and services, non-functional, e.g., lexica, corpora, etc., as well as all the related entities e.g., persons, organizations, licences, and projects. To better support interoperability some linked ontologies are going to be related to widespread approaches, such as DCAT[7], FOAF[8], and Schema.org[9].

The semantic annotations in ELG-SHARE are marked as mandatory, recommended, or optional, guiding this way the user which information is required, and which would be desirable. We note that each LT resource is also associated with the respective license and terms of use for its distribution. Also, for certain fields, designers choose to either extend or restrict the accepted value data types, for example a built-in primitive `xs:string` can be restricted to an enumeration of predefined strings, to increase ease of usage and minimize ambiguity. Also, the schema is hosted on a web-based interface, along with corresponding documentation [4].

Having described the motivation and related work, in the following sections we describe the general design of the AI4EU conceptual model and its interpretation as a Semantic Web ontology.

3 The AI4EU Conceptual Semantic Model

3.1 Model Design

The central concept of the model is naturally the *AI Resource*. This can be specialized by various subclasses that conceptualise different types of technology e.g., Dataset, Software, Hardware, Ontology, etc. A single resource can potentially have more than one *Distribution*, depending on the packaging, licensing, access mechanisms, and other parameters. Additionally, each distribution is foreseen to be accompanied by its *Documentation*, which can also entail different objects with different characteristics (e.g., Wikis, Text Documents, Manuals, Code Tests and

[6] https://www.european-language-grid.eu/

[7] https://www.w3.org/ns/dcat

[8] http://xmlns.com/foaf/spec/

[9] https://schema.org

Examples, etc.). Such different types of documentation can also be modelled as specialisations of the generic Documentation class.

Furthermore, an AI Resource is associated with different descriptive entities that help characterise the resource, i.e., the topic(s) it refers to, relevant keywords, and the computational resources to which it mainly relies for functioning.

Another aspect of a resource is its usage on applications, services, products and challenges. These are modelled through the *Application* class that will entail the properties that describe it (name, URL and so on), and is also connected to an *Application Area*.

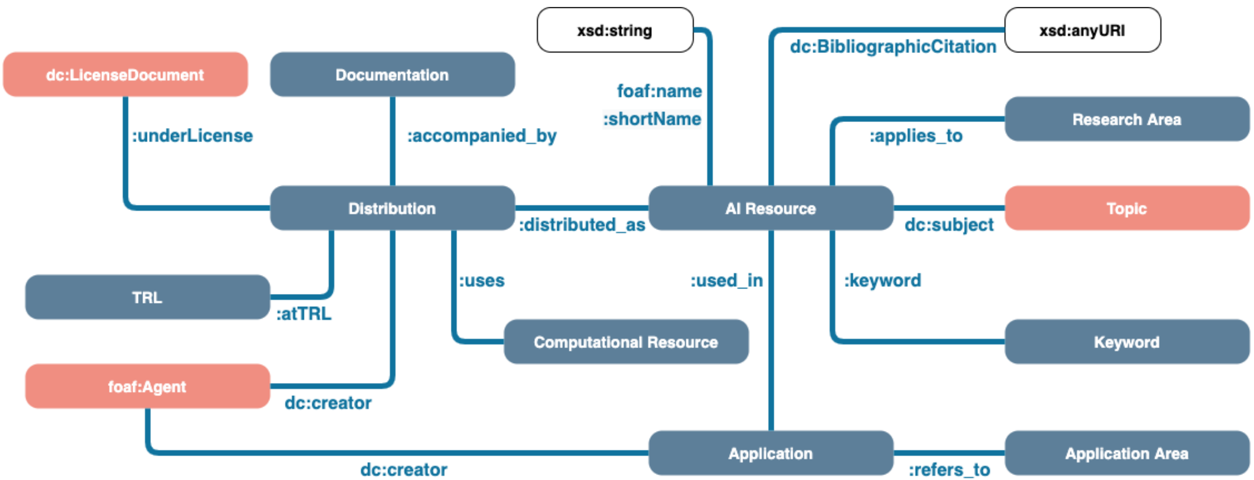

Figure 1: AI4EU high-level conceptual model

3.2 Model Definition Using Semantic Web Technologies

The representation of the presented conceptual model in a machine-readable formalisation uses the Web Ontology Language (OWL) W3C Specification[10]. In accordance with the reuse principles encouraged by the Semantic Web and Linked Data paradigms, the ontology imports and uses constructs from already defined and widely used ontologies and vocabularies. Specifically, the AI4EU ontology uses:

- Dublin Core Terms[11] for basic resource information (e.g., title, comment)
- FOAF for basic person and organisational information
- The Computer Science Ontology[12] for the taxonomy of computer science domains/topics to which a resource may pertain

Also, to design, extend, and maintain the AI4EU semantic model, a particular methodology has been compiled. This process can be summarised by the following workflow:

[10] https://www.w3.org/OWL/

[11] http://purl.org/dc/terms/

[12] https://cso.kmi.open.ac.uk/topics/

- Utilization of Protégé[13] or WebProtégé[14] for editing the ontology, which is common practice for ontologists.
- Versioning and collaboration processes are carried out using GitHub. Ontologists push their local edits to the relevant repository at their discretion, when substantial changes are added into the ontology.
- The WP3 task leader group is notified on the updates when major changes have been applied, in order to initiate an evaluation cycle (as described in the relevant section of this report).

The namespace for the AI4EU ontology is defined as: http://www.ai4eu.eu/ontologies/core#, while the prefix `ai:` was habitually used for abbreviating the namespace and should be treated as canonical. The following figure showcases the main classes and properties defined by and used in the AI4EU ontology.

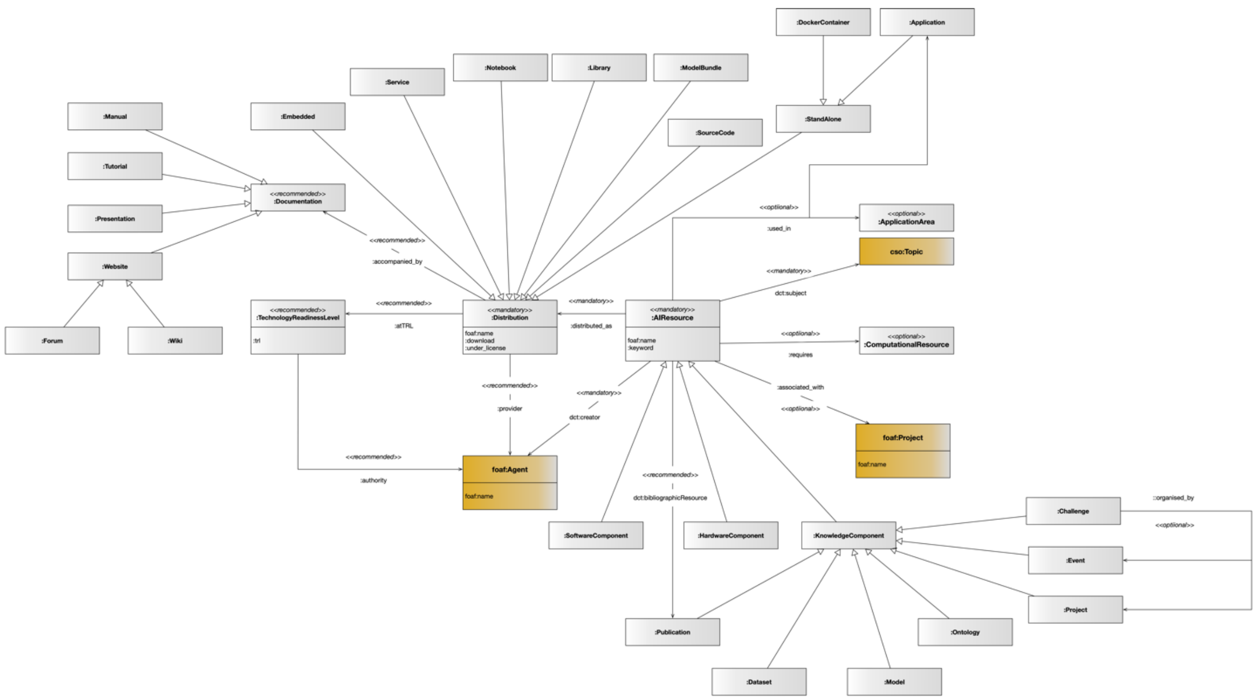

Figure 2: UML representation of the core AI4EU ontology concepts

In addition to the usage of the Computer Science Ontology for determining the scientific scope of a resource two controlled vocabularies were designed for formalizing references to Research Areas and Application Areas relevant to a resource. The vocabularies were described using the widely used Simple Knowledge Organization System (SKOS)[15] W3C Recommendation. The categories and areas included in the vocabularies have been discussed and agreed on by representatives of WP2, WP3, and WP7.

As the conceptual schema defines relations between all the core entities, a query answering system built on top of the schema will be able to navigate it starting arbitrarily from any class.

---

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

[14] https://protege.stanford.edu/products.php#web-protege

[15] https://www.w3.org/2009/08/skos-reference/skos.html

Thus, the ontology supports queries focusing on resources, topics, application areas, people, requirements as all these factors can be the main interest of a user seeking to find information and resources. Some indicative queries showcasing the ability to answer queries of a different focus are presented in the annex of this report.

As a last step of the ontology maintenance process, when a version of the ontology is tagged as final, it is published online and made accessible from a persistent URL.

We now present an assessment and comparison with the semantic descriptions of the other major platforms and the resources offered therein.

3.3 Comparison with Models Used in Relevant Platforms

BEAT: The BEAT platform focuses its use mainly for reproducing experimental results and re-using algorithms and data for benchmarking and comparisons. In the case of AI4EU, we treat AI Resources as components with possibly multiple areas of application, including experiments and bibliographic references, and therefore able to be incorporated as building blocks in various application types, either research-, educational-, or commercial/industrial-purpose systems. Although BEAT provides an API for access by third-party applications, its focus is not on importing/exporting data or algorithms for deployment in third-party infrastructure. Additionally, AI4EU includes discussions, networking possibilities, as well as rich content for new users to educate, discover and finally incorporate AI resources into their own processes. Thus, since an AI Resource may be referenced or incorporated in processes of different nature, a well-defined data model is required, that provides appropriate semantic descriptions for interoperability purposes, regardless of the resource implementation and execution details. Concerning the format of the data exchanged between the resources, this can be available in the form of protocol buffers, an interface description language that describes the exchanged data structure, as in the case of ACUMOS [5] and used for the onboarding process in AI4EU[16].

Bonseyes: By the description of the project's goals, BONSEYES resources appear to be a subset of the AI4EU resources that can be well described by the semantic model that we present in this deliverable. Fields for datasets, machine learning modules, and hardware components are readily supported, and, additionally, forums and discussions, wiki-pages, scientific publications and so on. Thus, the difference from the BONSEYES platform is that in AI4EU, apart from establishing a large and up-to-date repository of datasets, AI resources, components, social activity and discoverability are also promoted by faceted search, respective user groups, discussions, and networking that is driven by the interest in AI technology.

European Language Grid: The metadata model approach of ELG is quite similar with the one utilized by AI4EU, though it has a narrower orientation mainly towards language technologies. In AI4EU, AI resources can not only include language technology-related tools and datasets, but a variety of components, e.g., publications, software components, pre-trained ML models, etc. that can be used for a variety of use-cases and applications. In the AI4EU semantic model we also distinguish among mandatory, recommended, and optional fields, since we need, on the one hand a simple publication process, and on the other, a well described repository with enough metadata that are employed by searching and possibly automated reasoning procedures. In our approach we too link the FOAF data model, as well as OWL and SKOS among others.

---

[16] AI4EU Deliverable D2.7, Interoperability design and implementation choices reference document

Table 1: Entities covered by the ontologies used in each R&D project

Entities	BEAT	ELG	Bonseyes	AI4EU
Users and User Types	✓	✓	✓	✓
Datasets	✓	✓	✓	✓
Software Components	✓	LT-related only	✓	✓
Hardware Components	-	-		✓
Knowledge Components	-	✓		✓
Challenge			✓	✓
Application Area	-	✓		✓
Distribution	-	-		✓
Documentation	✓	✓		✓
Links to external ontologies	-	✓		✓

4 The AI4EU Knowledge Graph

This section provides details on the design and deployment of the knowledge graph based on the AI4EU ontology and walks through examples for the main operations supported by the knowledge graph deployment environment.

4.1 Design and Deployment

The AI4EU Knowledge Graph is a Semantic Web graph specified using RDF and RDFS. RDF is the Resource Description Framework formalism defined by W3C which is the basis of the Semantic Web. RDFS is RDF Schema, a simple ontology specification language for RDF graphs.

Design: RDFS enables users to define class hierarchies with possibly multiple inheritance. It also enables users to declare properties and their signature, called domain and range. The domain is the type of the subjects of the property and the range is the type of the values of the properties. Domain and range enable the inference engine to infer the type of such resources. Hence, domain and range are not meant to be used for type checking. Type checking can be performed using SHACL[17], a language that allows the validation of RDF graphs against complex conditions.

Properties can be arranged into hierarchies of property types. As already mentioned, the AI4EU ontology reuses several existing vocabularies such as Dublin Core, the Computer Science Ontology, DBPedia URI for countries, the FOAF vocabulary, etc.

An RDF graph is a labelled oriented multigraph made of edges, called triples, and nodes. Nodes represent resources and literal values which are constants. Literal values are typed using XML schema datatypes such as integer, string and date. Edges of the graph, so called triples, represent properties. RDF resources are assigned to types using classes, which are in turn defined using RDFS. Properties are typed using property signature definition in RDFS.

[17] https://www.w3.org/TR/shacl/

RDF and RDFS graphs are specified in textual format with several possible syntax: RDF/XML, Turtle, JSON LD, etc. In our case, we use the Turtle format to ensure readability and, to an extent, forward compatibility as initial tools supporting the emerging RDF 1.2 specification are at the moment able to handle only Turtle. The example in Figure 3 shows an extract of an AI resource which describes the Ellogon linguistic platform, with the corresponding diagrammatic representation right below.

```
:ellogon a ai:SoftwareComponent ;
  foaf:name "Ellogon"^^xsd:string ;
  ai:distributed_as :ellogon-distro ;
  ai:requires :CPU ;
  dcterms:subject cso:natural_language_processing_systems ;
  ai:keyword "Linguistic Pipelines"@en ,
           "Language Engineering Platform"@en ,
           "NLP"@en ;
  dcterms:bibliographicCitation "https://arxiv.org/pdf/cs/0205017"^^xsd:anyURI ;
  dcterms:creator :ncsr .
:ellogon-distro: a ai:StandAlone ;
                   ai:atTRL :trl9 ;
                   ai:accompanied_by :ellogon-devguide ,
                                     :ellogon-specs ,
                                     :ellogon-userguide ;
                   ai:under_license :lgpl ;
                   ai:download
"http://www.ellogon.org/index.php/download"^^xsd:anyURI ;
                   ai:language_used <http://dbpedia.org/resource/Tcl> .
:ncsr a owl:NamedIndividual, foaf:Organization ;
        foaf:name "National Centre for Scientific Research - Demokritos"@en ;
        foaf:homepage <http://www.demokritos.gr> ;
        ai:location <http://dbpedia.org/resource/Greece> .
```

Figure 3: Example of RDF metadata

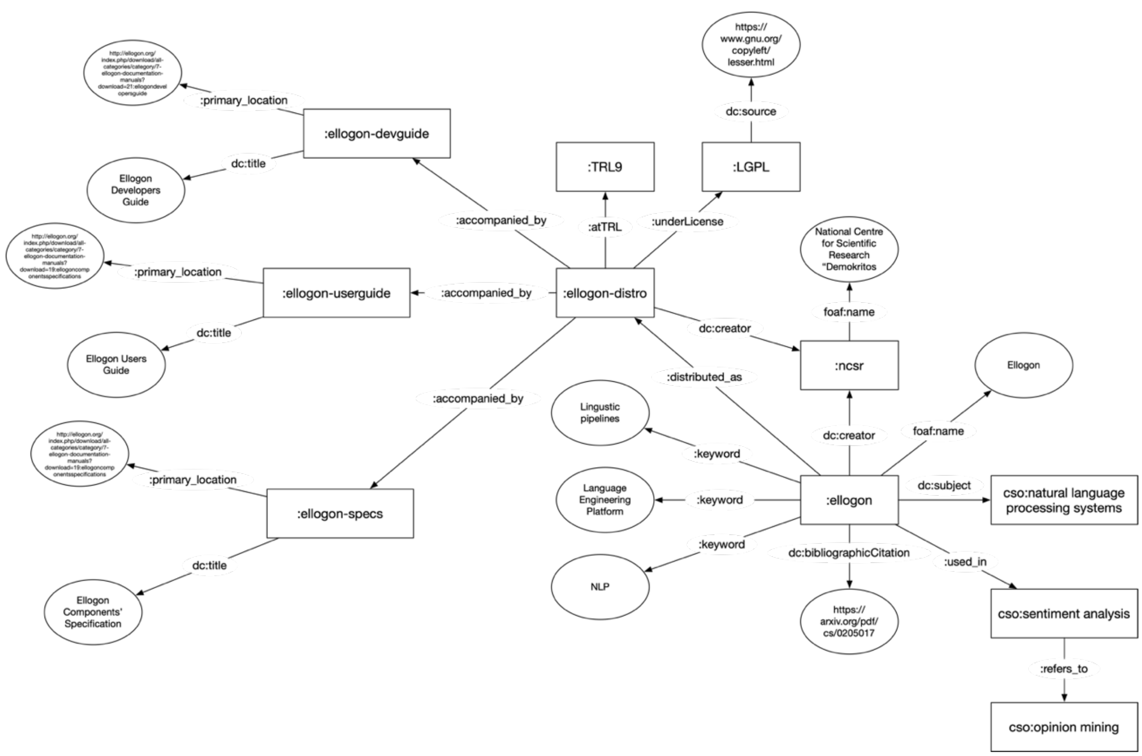

Figure 4: Graph representation of RDF metadata example

The AI4EU ontology defines and uses the namespace: http://www.ai4eu.eu/ontologies/core# which means that all class and property names starts with this URI. For example, the class `SoftwareComponent`, shown below in Figure 5, has the following complete name: http://www.ai4eu.eu/ontologies/core#SoftwareComponent.

```
:SoftwareComponent rdf:type owl:Class ;
      rdfs:subClassOf :AIResource ;
      rdfs:comment "A programmatic AI resource."@en ;
      rdfs:label "Software Component"@en .
```

Figure 5: Example of an OWL class definition

Similarly, the example below showcases the definition of `ai:location`, an OWL property with `foaf:Organization` as its domain.

```
:location rdf:type owl:ObjectProperty ;
          rdfs:domain foaf:Organization ;
          rdfs:comment "The country where an organisation participating in the
development or management of a resource is located. It is recommended that the DBPedia
URI of the corresponding country is provided."@en ;
          rdfs:label "Location"@en .
```

Figure 6: Example of an OWL property definition

In addition to the definition models, we have defined SHACL shapes to validate RDF metadata and the ontology. An exemplary shape for validating the RDF graph is presented in the following figure. Note that instances of `ai:AIResource` ***must*** have at least one `creator` property with a value of type Organization, one property `name`, and the `subject` property must take its value in the Computer Science Ontology namespace.

```
ai:resourceShape a sh:NodeShape ;
sh:targetClass ai:AIResource ;
sh:property [
    sh:path  dct:creator  ;
    sh:minCount 1 ;
    sh:class foaf:Organization ] ;
sh:property [
    sh:path  foaf:name  ;
    sh:minCount 1 ] ;
sh:property [
    sh:path  dct:subject  ;
    sh:minCount 1 ;
    sh:pattern "^https://cso.kmi.open.ac.uk/topics/" ] ;
```

Figure 7: Example of a SHACL space definition

Deployment: The deployment of the AI4EU Knowledge Graph is done using a Semantic Web tool called “triple store”. A triple store is a graph database that implements RDF, RDFS and the SPARQL query language. It may provide entailments according to the ontology: class subsumption, property signature, etc. A triple store also provides a SPARQL endpoint which means that the Knowledge Graph is accessible on the Web (and the Internet) and it can be queried and navigated via the HTTP protocol.

Several formats are available to access the graph. Examples include, Turtle, RDF/XML, JSON-LD, etc. A SPARQL endpoint in particular, may also provide SPARQL Update to load RDF documents, modify and update the RDF graph. Hence, it is possible to enrich the RDF graph by means of SPARQL Update queries.

As the AI4EU Knowledge Graph is specified with the W3C Semantic Web standards, it can be uploaded to any triple store. Hence, we have a large degree of freedom for the deployment. Examples of triple stores are: Virtuoso[18], Jena[19], Corese[20], etc.

4.2 Tools and Interfaces

Protégé can be used to edit the ontology and the RDF metadata. The figure below shows an extract of the AI4EU ontology class hierarchy in the Protégé editor.

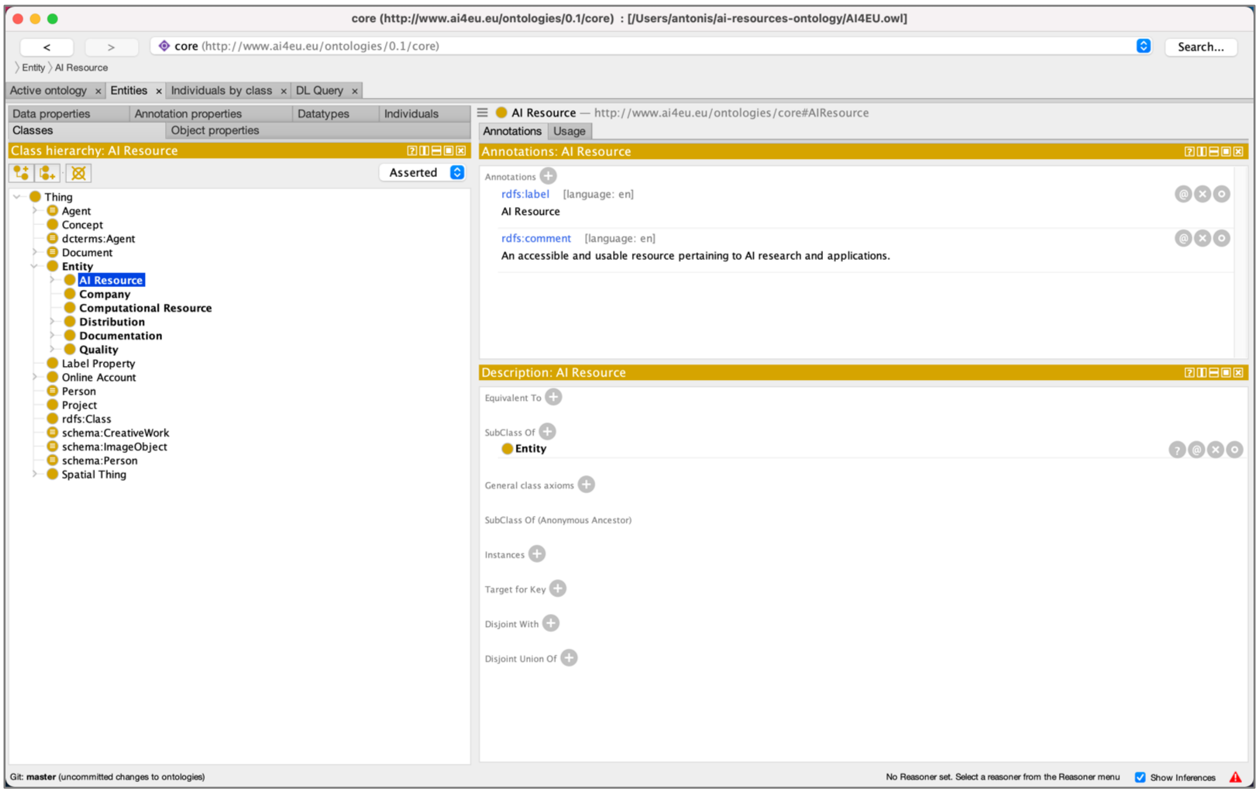

Figure 8: The AI4EU ontology loaded in Protégé

A SPARQL endpoint prototype is set up to provide access to AI4EU Knowledge Graph using Inria Corese Semantic Web Factory: http://corese.inria.fr/srv/service/ai4eu.

The actual prototype provides a triple store accessible by means of a SPARQL endpoint. It offers RDF, RDFS, OWL RL, SHACL, as well as SPARQL Query & Update. The SPARQL endpoint is provided with a user interface for SPARQL queries with a set of predefined queries. The queries are presented in a text editor and, hence, users can modify them. Some query parameters can be chosen through the interface such as type of resource, keyword, etc.

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

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

[20] https://project.inria.fr/corese/

Query results are displayed with hypertext links to resource URI which allow hypertext navigation in the AI4EU Knowledge Graph. The example below shows the user interface with a query that searches for resources with the keyword "NLP". In the example, there is one answer: "Ellogon". The interface proposes three select menus which are generated from the content of Knowledge Graph. The "Type" menu is fed with types of AI resources such as "SoftwareComponent". The "Keyword" menu is fed with possible values for the "keyword" property and the "Name" menu is fed with names of existing resources.

In addition to the SPARQL endpoint, the server provides specific graphic display for Semantic Web ontologies. In particular, the class hierarchy can be displayed in a graphic mode. When a query result contains information about locations, e.g., the country where a resource comes from, it is possible to display the locations on a map. The server has also available specific graphic display for queries with aggregate operations such as "count".

4.3 Example: Describing and injecting an AI Resource

Resource injection is accomplished through three different means:

i. Predefined RDF documents can be loaded when the server starts.
ii. RDF documents can be loaded using the `load` SPARQL update statement. The `load` statement accepts as argument a URI that is the URL of an RDF document. The content of the document is then parsed and loaded in the triple store. A usage example of the statement is: `load <http://example.org/ai4eu/ellogon.ttl>`
iii. RDF metadata can be inserted using the insert data SPARQL update statement. For example:

```
insert data {
 :corese a ai:SoftwareComponent ;
  foaf:name "Corese"^^xsd:string ;
  ai:distributed_as :corese-distri ;
  ai:requires :CPU ;
  ai:keyword "Semantic Web"@en ,
           "Linked Data"@en ,
           "Knowledge Graph"@en ;
  dct:subject https://cso.kmi.open.ac.uk/topics/semantic_web ;
  dct:creator :inria .
:inria a foaf:Organization ;
        foaf:name "INRIA"@en ;
        foaf:homepage "http://www.inria.fr"^^xsd:anyURI ;
        ai:location http://dbpedia.org/resource/France .
 }
```

Figure 9: insert data statement example

Additionally, resource descriptions can be updated using the `delete-insert-where` SPARQL update statement, e.g.:

```
Delete { ?x foaf:name "INRIA"@en }
insert { ?x foaf:name "Inria"@en }
where { ?x foaf:name "INRIA"@en }
```

Figure 10: delete-insert-where statement example

4.4 Example: Searching for AI Resources

Resources are searched using the SPARQL query language. In the example below, we look to retrieve the name, keywords, creator and location of resources having the `SoftwareComponent` type. In addition, we query the DBPedia remote SPARQL endpoint using the `service` clause, in order to get the coordinates of the location. Coordinates information allows us to display the location on a map, using e.g., Open Street maps.

```
prefix ai:     <http://www.ai4eu.eu/ontologies/core#>
prefix geo:  <http://www.w3.org/2003/01/geo/wgs84_pos#>
prefix dct:   <http://purl.org/dc/terms/> .
select ?a ?name ?type  (group_concat(distinct ?w ; separator = "; ") as ?ww) ?location
where {
  ?a a ai:SoftwareComponent  ;
  foaf:name ?name ;
  ai:keyword ?w ;
  dct:creator ?org .
  optional {
    ?org ai:location ?location
        service <http://dbpedia.org/sparql> {
            ?location geo:lat ?lat ; geo:long ?lon
        }
    }
}
group by ?a
```

Figure 11: Using SPARQL queries to search for resources

The query results retrieved are presented below.

Result

	a	location	name	type	ww
1	<http://www.ai4eu.eu/ontologies/data#Sentinel_Datasets>	dbe:Brussels	"Sentinel Datasets"@en	ai:SoftwareComponent	"Image data; Satellite Data"@en
2	<http://www.ai4eu.eu/ontologies/data#corese>	dbe:France	"Corese"	ai:SoftwareComponent	"Semantic Web; Linked Data; Knowledge Graph"@en
3	<http://www.ai4eu.eu/ontologies/data#drools>	dbe:United_States	"DROOLS"	ai:SoftwareComponent	"Rules Management System"@en
4	<http://www.ai4eu.eu/ontologies/data#ellogon>	dbe:Greece	"Ellogon"	ai:SoftwareComponent	"Linguistic Pipelines; Language Engineering Platform; NLP"@en
5	<http://www.ai4eu.eu/ontologies/data#mc-giver>	dbe:Italy	"mc-giver"	ai:SoftwareComponent	"CNN; Visual Document; Classification"@en
6	<http://www.ai4eu.eu/ontologies/data#sciff>	dbe:Italy	"SCIFF"	ai:SoftwareComponent	"Logic Programming; Abductive Logic Programming; Monitoring"@en
7	<http://www.ai4eu.eu/ontologies/data#sunnyas>	dbe:Italy	"SUNNY-AS"	ai:SoftwareComponent	"Algorithm Selection; kNN; Portfolio-based optimization; ML"@en
8	<http://www.ai4eu.eu/ontologies/data#unbbayes>	dbe:Brazil	"UnBBayes"@en	ai:SoftwareComponent	"Bayesian Network; Probabilistic Graphical Model; Probabilistic Reasoning"@en
9	<http://www.ai4eu.eu/ontologies/data#wildcatpytorch>	dbe:France	"wildcat.pytorch"	ai:SoftwareComponent	"Object Segmentation"@en

Figure 12: Results of a SPARQL query with hypertext links

As we can observe, the resulting URIs are displayed as hypertext links. If we click on the “ellogon” hypertext link, we obtain the answer shown below which is the RDF metadata description of the resource. Hence, it is possible to navigate the RDF metadata graph as a hypertext, which is another means for discovering AI resources.

	subject	property	object
1	<http://www.ai4eu.eu/ontologies/data#ellogon>	dc:bibliographicCitation	"https://arxiv.org/pdf/cs/0205017"^^xsd:anyURI
2	<http://www.ai4eu.eu/ontologies/data#ellogon>	dc:creator	<http://www.ai4eu.eu/ontologies/data#ncsr>
3	<http://www.ai4eu.eu/ontologies/data#ellogon>	dc:subject	<https://cso.kmi.open.ac.uk/topics/natural_language_processing_systems>
4	<http://www.ai4eu.eu/ontologies/data#ellogon>	ai:distributed_as	<http://www.ai4eu.eu/ontologies/data#ellogon-distro>
5	<http://www.ai4eu.eu/ontologies/data#ellogon>	ai:keyword	"Language Engineering Platform"@en
6	<http://www.ai4eu.eu/ontologies/data#ellogon>	ai:keyword	"Linguistic Pipelines"@en
7	<http://www.ai4eu.eu/ontologies/data#ellogon>	ai:keyword	"NLP"@en
8	<http://www.ai4eu.eu/ontologies/data#ellogon>	ai:requires	<http://www.ai4eu.eu/ontologies/data#CPU>
9	<http://www.ai4eu.eu/ontologies/data#ellogon>	rdf:type	ai:AIResource
10	<http://www.ai4eu.eu/ontologies/data#ellogon>	rdf:type	ai:Entity
11	<http://www.ai4eu.eu/ontologies/data#ellogon>	rdf:type	ai:SoftwareComponent
12	<http://www.ai4eu.eu/ontologies/data#ellogon>	rdf:type	owl:NamedIndividual
13	<http://www.ai4eu.eu/ontologies/data#ellogon>	rdf:type	owl:Thing
14	<http://www.ai4eu.eu/ontologies/data#ellogon>	owl:sameAs	<http://www.ai4eu.eu/ontologies/data#ellogon>
15	<http://www.ai4eu.eu/ontologies/data#ellogon>	foaf:name	"Ellogon"

Figure 13: Results of hypertext navigation for the Ellogon resource

4.5 Example: Visualisation of AI Resource Information

The Web server provides graphic display with openstreetmap[21] for locations. In this example case, we search the latitude and longitude on DBpedia SPARQL endpoint[22]. Below, the display of such locations for the query above is shown.

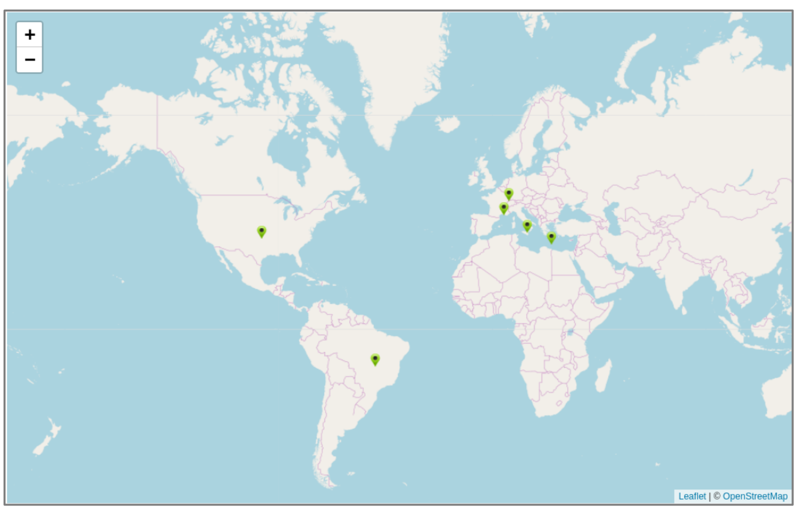

Figure 14: Map of location of resources

21 https://www.openstreetmap.org/

22 http://dbpedia.org/sparql

Furthermore, the Web server provides visual representations for aggregate queries. The example below presents the occurrences of keywords in the RDF graph.

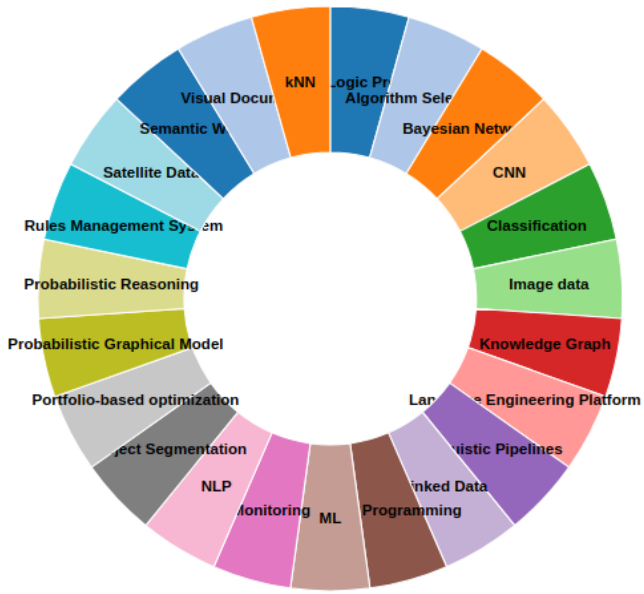

Figure 15: Pie chart of keywords found in the RDF graph

The class hierarchy of the AI4EU ontology can be also visualised. An extract is presented below.

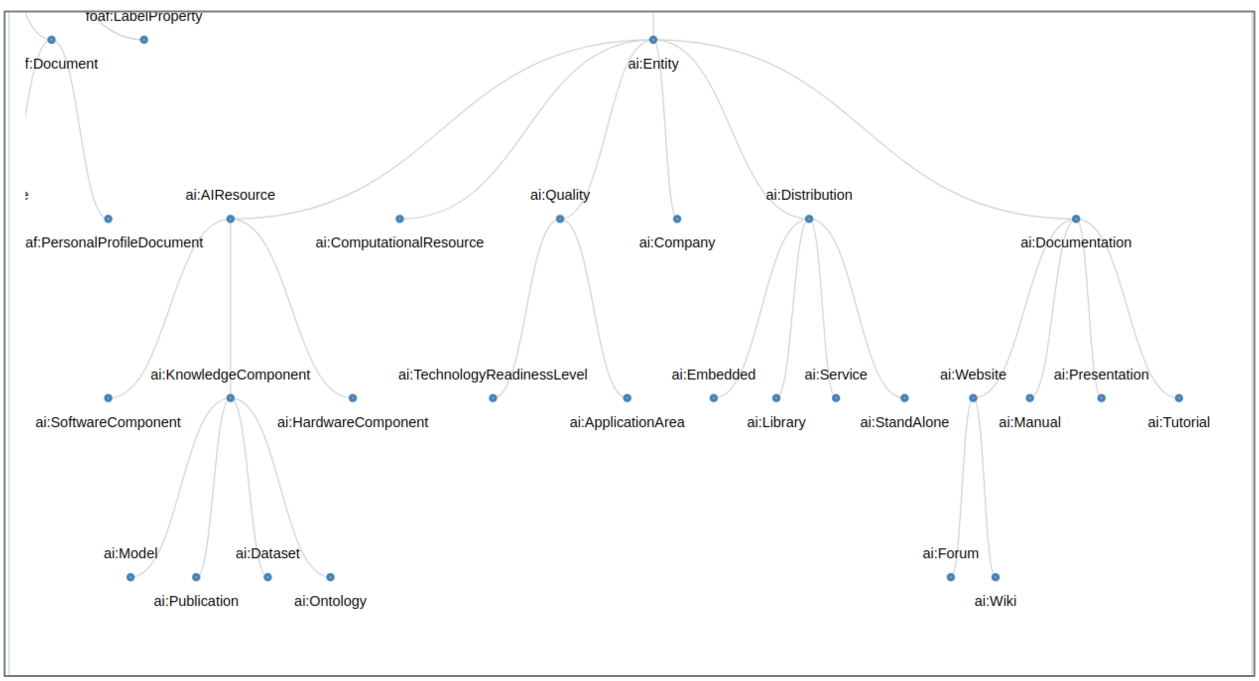

Figure 16: AI4EU class hierarchy as presented from the hosting server

Finally, the Web server exposes an RDF validation service, using the defined SHACL shapes, both for validating RDF metadata[23] and the AI4EU ontology[24]. An exemplary validation report is presented in the figure below, indicating that two resources miss a `dc:subject` property.

```
prefix rs: <http://www.w3.org/2001/sw/DataAccess/tests/result-set#>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix skos: <http://www.w3.org/2004/02/skos/core#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix ft: <http://ns.inria.fr/sparql-template/format/ds/>
prefix sh: <http://www.w3.org/ns/shacl#>
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix dbe: <http://dbpedia.org/resource/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix doc: <http://w3c.github.io/data-shapes/shacl/#>
prefix dc: <http://purl.org/dc/terms/>

[a sh:ValidationReport ;
  sh:conforms false ;
  sh:result <urn:uuid:53abf3f6-5ff3-401a-a2c5-eeba7f68a9fe> ;
  sh:result <urn:uuid:a3051c37-3acf-44db-abb0-a32ac437f315> ;
  sh:result <urn:uuid:a6e88a36-5cf4-42bd-8e7b-026e6cf55232> ;
  sh:result <urn:uuid:f5259c5c-9482-4a81-b35b-ab1f8cec45bc>] .

<urn:uuid:53abf3f6-5ff3-401a-a2c5-eeba7f68a9fe>
  a sh:ValidationResult ;
  sh:focusNode <http://www.ai4eu.eu/ontologies/data#unbbayes> ;
  sh:resultMessage "Fail at: [sh:minCount 1 ;
  sh:path dc:subject ;
  sh:pattern \"^https://cso.kmi.open.ac.uk/topics/\"]" ;
  sh:resultPath dc:subject ;
  sh:resultSeverity sh:Violation ;
  sh:sourceConstraintComponent sh:MinCountConstraintComponent ;
  sh:sourceShape _:b9216 ;
  sh:value 0 .

<urn:uuid:a3051c37-3acf-44db-abb0-a32ac437f315>
  a sh:ValidationResult ;
  sh:focusNode <http://www.ai4eu.eu/ontologies/data#Sentinel_Datasets> ;
  sh:resultMessage "Fail at: [sh:minCount 1 ;
  sh:path dc:subject ;
  sh:pattern \"^https://cso.kmi.open.ac.uk/topics/\"]" ;
  sh:resultPath dc:subject ;
  sh:resultSeverity sh:Violation ;
  sh:sourceConstraintComponent sh:MinCountConstraintComponent ;
  sh:sourceShape _:b9216 ;
  sh:value 0 .
```

Figure 17: SHACL shape validation report

5 Model Evaluation

The evaluation of the model relied on two evaluation processes:

- Consistency evaluation: the process aims to identify consistency errors in the ontology. It is performed after each ontology release using the appropriate SHACL shapes that have been defined by the ontology development working group. As more descriptions are becoming available, the shape set is enriched with additional rules and the evaluation becomes further

[23] http://corese.inria.fr/srv/service/ai4eushape?mode=/data/demo/ai4eu/shape/data.ttl

[24] http://corese.inria.fr/srv/service/ai4eushape?mode=/data/demo/ai4eu/shape/ontology.ttl

refined. Following the production of the SHACL validation report, the ontology is updated in order to accommodate the validation errors discovered.

- Expressivity evaluation: the second process has to do with the adequacy of the ontology to express the targeted AI resources in sufficient detail and with sufficient clarity. To assess this, a two-stage process was followed: members of the AI4EU consortium were provided with a spreadsheet template and asked to describe AI resources available or known to them. Subsequently, the ontology development team expressed the provided descriptions in RDF and under the schema defined by the ontology, taking into account discrepancies in terminology and remarks provided by the testers, in terms of what was unclear, what was missing, etc. The ontology was then updated in order to cover descriptive aspects that were unclear or omitted from the previous version.

6 The AI4EU Data Model

One of the main goals of AI4EU is to integrate well with other platforms. In order to exchange AI Resources bidirectionally between platforms, a model is needed to describe the metadata. This *Metadata Model* or *Core Data Model* is derived from the ontology and represented as relational data model. It was agreed that the core data model should be the least common denominator of the connected platforms to keep it as simple as possible. So, the core data model will not be a superset with all attributes of all platforms merged in, but a subset with mandatory attributes that make up a basic catalogue entry. Having a simple data model will make it for everybody easier to integrate. The current version of the model has already been aligned (green outline) with the platforms Drupal Repository, Search, AI4EU Experiments (AI4EU components, in white), ELG and Beat (third-party platforms, in grey), with other platforms to follow in the future.

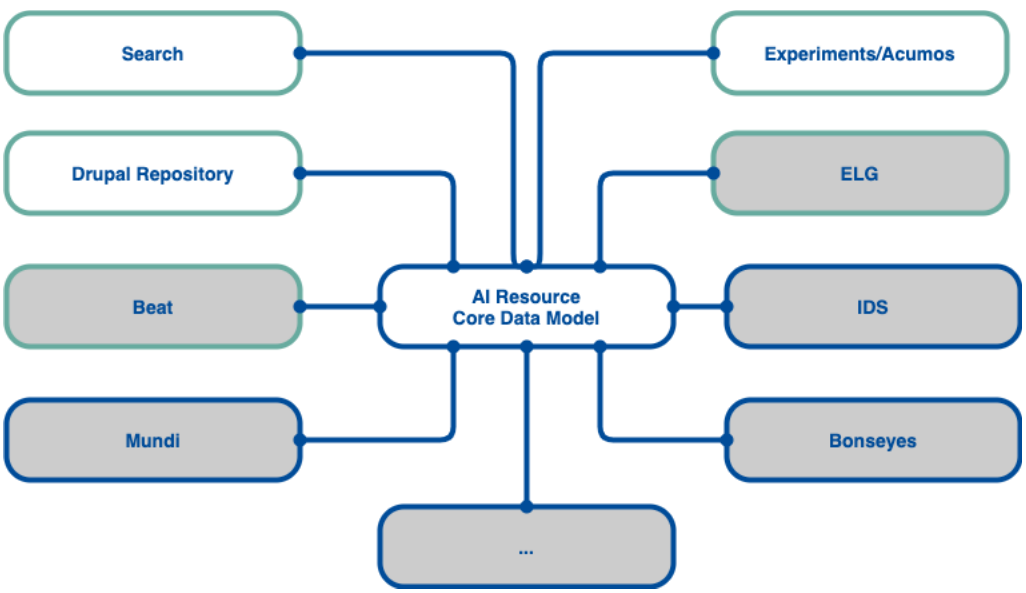

Figure 18: Bi-directional mapping between core data model and connected platforms

The model has a number of categorical entities, the values of which are listed below.

Distribution Type: can be docker-container, notebook, native-executable, zip, java-archive

Content Type: can be any of internet mime-types like text/plain, text/csv, application/pdf, audio/mp3, video/h.264

Support Type: can be free, discussion-forum, commercial

Furthermore, and in accordance with the conceptual model's vocabularies, the data model accepts the values included in the following table for AI Research Categories and **AI Application Areas**.

Table 2: Core data model: AI research categories and application areas

AI Research Categories	**AI Application Areas**
Explicable AI	AI for agriculture
Verifiable AI	AI in health
Physical AI	AI for citizen services & education
Integrative AI	AI for robotics
Collaborative AI	AI for industry and manufacturing
Algorithm selection	AI in autonomous driving and mobility
Computational logic	AI for art and music
Computer Vision	AI for environment and sustainability
Constraints and SAT	AI for IoT
Decision support systems	AI for cybersecurity
Heuristic search and game playing	AI for media
Knowledge representation and reasoning	AI for telecommunication
Machine learning	AI for finance & insurance
Multi-agent systems	AI for law
Planning	AI in retail and ecommerce
Speech and signal processing	AI in software engineering
Natural language and dialogue processing	AI in Human Resources
Probabilistic models	Trusted and privacy preserving AI
	AI for Ambient Intelligence

The following diagram showcases the entities and their relations. It should be noted that all relations are one-to-many so, for example, one AI Resource may have many documents, distributions or AI Application Areas associated with it.

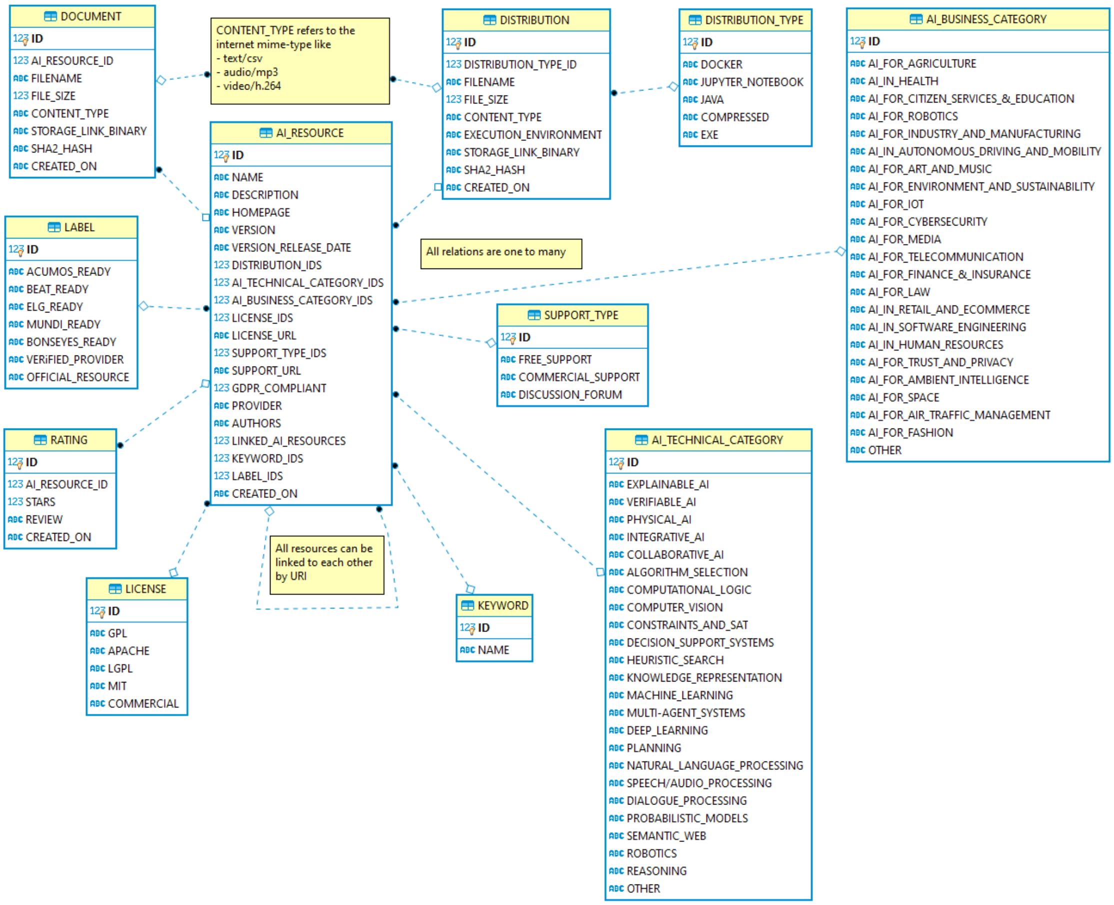

Figure 19: AI4EU Core Data Model

- **AI Resource**: This is the central entity which holds the main attributes of an AI Resource. AI Resources can be linked together for example to show the connection between a trained neural network model and the dataset that has been used for training.
- **Document**: Can be any type of document or picture describing the AI Resource or its use. Typically .doc or .pdf or .png.
- **Rating**: The rating contains a user review consisting of a review text and number of stars with the more stars the better
- **Label**: The label is a informs about on which of the connected platforms it can be imported or executed
- **License**: Contains the license details
- **Keyword**: Those are free text words that a provider can add to the resource to highlight features or topics.
- **Distribution**: This entity describes the execution environment and maybe points to executables or other binary files.

7 Alignment Process: AI4EU and ELG

The purpose of the joint experiment with ELG representatives is to establish the baseline for integrating a third-party, domain-centred knowledge base with AI4EU. The characteristics taken into account when designing the experiment are:

1. AI4EU, as a knowledge base of broader scope, cannot and should not focus on domain-specific technical and scientific characteristics of the resources described in it. It addresses the needs of a wider audience and thus should provide adequate details for understanding the purpose and applicability of resources but details relevant to the domain expert must remain accessible at the domain-specific knowledge bases. AI4EU must, however, establish the mechanisms for moving to the relevant resource in the domain-specific databases.
2. The AI4EU data model provides an import/export format that allows partner platforms to describe available resources and link back users to their original location. Partner platforms should implement the relevant entry points in their web addresses allowing for automated scanning (and de-duplication) from the AI4EU platform, which should be registered for periodic scans. They may optionally implement a push service to notify the AI4EU central catalogue of new resource additions and eventual removals. No peer-reviewing is implemented on the AI4EU platform side, except for special tracks (e.g., for a conference).

At this stage, the integration process focuses on the mapping at the schema level of AI4EU and ELG, i.e., to define correspondences between the schemas informing the two knowledge bases. It was decided to initiate the mapping process, described in Section 2.3, using the Minimal Descriptor Set of ELG, in order to preserve the focus of AI4EU at a relatively domain-agnostic level while sufficiently describing the resource so that an interested user can understand it and possibly seek further information in the ELG database.

7.1 Mapping process

To proceed with the mapping task, a shared collaboration space was set up and operated using Google Drive folders for maintaining information, minutes, and working documents. The core working document is a spreadsheet on which AI4EU and ELG produced the correspondences between AI4EU data model concepts and the ELG set of mandatory descriptors. An iterative process of mapping suggestions from ELG, confirmation or clarification from AI4EU, and finalisation of a mapping was carried out throughout the activity.

In addition to mapping the schemas, the relevant working groups of AI4EU and ELG proceeded to the mapping of the AI4EU resource categories (Research Categories and Application Areas) with the respective Resource Types used by ELG. In the case of Research Categories, as expected, a mapping applied only to categories meaningful for ELG (i.e., Language Processing). In contrast, most Application Areas were foreseen in both models.

7.2 Current Status and Next Steps

At this stage, we consider the produced set of mappings between the two schemas as finalised. The next step is the formal declaration of these mappings, in the form of including equivalence or partial equivalence relations in the two ontologies (`owl:sameAs`, `owl:equivalentClass`, `skos:ExactMatch`, `skos:closeMatch`, etc.). Both ontologies are not officially published and exposed via a persistent URI, so some parts of the process are subject to change as publishing decisions are discussed independently on both sides. However, tentative information on the URIs to be used and third-party ontologies to be used has been exchanged between the two collaborating teams, so that the formal definition of the mappings can start before the ontologies are officially published.

```
prefix ai: <http://www.ai4eu.eu/ontologies/core#>
prefix aicat: <http://www.ai4eu.eu/ontologies/ai4eu-categories#>
prefix ms: <http://w3id.org/meta-share/meta-share/>

…

ai:AIResource skos:broaderMatch ms:LanguageResource .
ai:version skos:exactMatch ms:version .
ai:keyword skos:exactMatch ms:keyword .
aicat:NLP            skos:broaderMatch            <http://w3id.org/meta-share/omtd-share/NaturalLanguageGeneration> .

…
```

Figure 20: Examples of AI4EU-ELG mappings

8 Potential Risks and Risk Management Measures

Given the described processes for updating the ontology and the core data model, as well as, connecting with external platforms, there are specific failure points that must be taken into consideration while continuing the development of the relevant AI4EU assets. Such points are:

- **Inconsistencies between the core data model and the ontology**: these could occur by changes on the user-facing requirements of the resource registration process. To mitigate the risk, the ontology development team is in close communication with the model development team and has access to the relevant repositories. Thus, it is able to monitor changes and discuss on the implications on the ontology before finalising any model evolution.
- **Discrepancies between the core data model and third-party models of platforms to be connected with AI4EU**: as discussed, such occurrences will in most cases be resolved by extending the core data model (and the ontology) to incorporate concepts that cover the observed conceptualisation. In cases where this isn't feasible, e.g., if the observed conceptualisation introduces inconsistencies in the model, the ontology design team will proceed to a partial redesign of the ontology and the schema of the core data model will be accordingly adapted.
- **Further recommendations**: this data model is subject to recommendations regarding privacy and ethics according to the general policy that AI4EU follows regarding this matter.

9 Conclusions

The document presents the rationale for designing the AI4EU conceptual model and implementing it as an OWL ontology. The model's design has gone through multiple evaluation steps in order to make sure that it carries sufficient expressive power without sacrificing its flexibility. Additionally, the extensive usage of already established schemas is expected to facilitate integration with existing and future knowledge bases and third-party platforms.

In the context of ontological work, a knowledge graph and an overlying search and visualisation service was built and deployed, showcasing the possibilities of using the knowledge graph for complex querying and inferencing.

Another important aspect of the conceptual model is its direct link with the produced metadata model, which forms the baseline for integrating, new or existing, domain-focused initiatives under the AI4EU data model. The establishment of a collection of core concepts, and the initial steps for defining an integration process will allow the AI4EU knowledge base to be closely linked with narrower communities and act as a central point that provides a sufficient overview of AI resources, while enabling easy access to finer-grained information available externally.

References

[1] Penny Labropoulou et al., "D2.3 – Metadata Schema", European Language Grid Deliverable, 2019.

[2] European Language Grid Metadata Model, 2019. Online: https://www.european-language-grid.eu/metadata/

[3] André Anjos, Laurent El Shafey, and Sebastian Marcel, "BEAT: An Open-source Web-based Open-science Platform, ArXiv, 2017.

[4] S. Zhao et al., "Packaging and Sharing Machine Learning Models via the Acumos AI Open Platform", in IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2018, pp. 841-846.

[5] Tim Llewellynn et al., "BONSEYES: Platform for Open Development of Systems of Artificial Intelligence" In Proceedings of the Computing Frontiers Conference (CF'17), New York, 2017, pp. 299-304

[6] Maksimov, Y V., Fricker, S., Tutschku, K. (2018) Artifact Compatibility for Enabling Collaboration in the Artificial Intelligence Ecosystem In: Lecture Notes in Business Information Processing (pp. 56-71). Springer Lecture Notes in Business Information Processing https://doi.org/10.1007/978-3-030-04840-2_5

Annex: Exemplary SPARQL queries covered by AI4EU

The annex presents some indicative queries useful for a search system built on top of a knowledge base informed by the AI4EU ontology and likely to be requested from users of the knowledge base.

Query 1: Retrieve all people with an expertise in Machine Learning

```
prefix ai:     <http://www.ai4eu.eu/ontologies/core#>
prefix foaf:   <http://xmlns.com/foaf/0.1//> .

select ?name where {
  ?a a foaf:Person ;
      foaf:name ?name ;
      ai:expertise <https://cso.kmi.open.ac.uk/topics/machine_learning> .
}
```

Query 2: Retrieve all computer vision resources

```
prefix ai:     <http://www.ai4eu.eu/ontologies/core#>
prefix dct:   <http://purl.org/dc/terms/> .

select ?resource ?name where {
  ?resource a ai:AIResource ;
      foaf:name ?name ;
      dct:subject < https://cso.kmi.open.ac.uk/topics/computer_vision> .

}
```

Query 3: Retrieve machine learning resources implemented in Java

```
prefix ai:     <http://www.ai4eu.eu/ontologies/core#>
prefix foaf:   <http://xmlns.com/foaf/0.1//> .

select ?resource ?name where {
  ?resource a ai:AIResource ;
      foaf:name ?name ;
      :distributed_as ?distro .
  ?distro a ai:Distribution ;
      ai:language_used <http://dbpedia.org/resource/Java_(programming_language)>.

}
```

Query 4: Retrieve computer vision resources applicable in Agriculture

```
prefix ai:     http://www.ai4eu.eu/ontologies/core#
prefix aicat: http://www.ai4eu.eu/ontologies/ai4eu-categories#
prefix dct:   <http://purl.org/dc/terms/> .

select ?resource ?name where {
  ?resource a ai:AIResource ;
      foaf:name ?name ;
      dct:subject < https://cso.kmi.open.ac.uk/topics/computer_vision> ;
      :used_in aicat:Agriculture .

}
```