EURIO: an ontology for publishing research projects' data

CORDIS

Everis

20/11/20

Publications Office of the European Union

Table of contents

1. Introduction 3
2. Terminology used 3
3. Context of use 4
4. Used vocabularies 4
5. Graphical representation 5
6. EURIO ..... **Error! Bookmark not defined.**
7. Concept evolution and versioning 20

1. Introduction

CORDIS is the portal responsible for publishing the results of EU-funded research and innovation projects. The research activities are carried out by a wide variety of organisations from several business sectors and scientific fields. The results of these projects are mainly new scientific knowledge (publications) and others kind of assets (e.g., tools, methodologies, data, etc.). All these results are relevant assets that should be reused to stimulate the economic activities of the related innovative sectors.

EURIO (EUropean Research Information Ontology) is the conceptual data model developed by the Publications Office of the European Commission that has been designed to represent and structure the CORDIS content in a semantic format improving its visibility, reusability and accessibility. The Ontology is built from a network of ontologies and reference data that allows to describe the administrative information of the research projects as well as its results, organisations and persons involved, among others.

The purpose of the Ontology is to be published as one of the already existing reference data assets managed by the Publications Office of the European Union. Reference data assets refer to ontologies, thesauri, taxonomies, authority tables, etc. Examples of such assets maintained by the Publications Office are EuroVoc thesaurus, EuroSciVoc taxonomy, ELI ontology, Language and Country authority lists. The complete list of assets can be found on EU Vocabularies Website[1].

The Ontology specified in this document is based on the OWL (Ontology Web Language) specification and is intended to facilitate the exchange of information. OWL is an RDF[2] vocabulary designed to facilitate interoperability between vocabularies (ontologies) published on the Web as Linked Open Data. Additional classes and attributes from other well-known vocabularies are re-used where necessary. Additionally, the Ontology also combines SKOS (Simple Knowledge Organisation System) properties that will allow to specify the types of some classes and attributes.

The work does not cover implementation issues like mechanisms to edit or publish data. However, there will be a public SPARQL endpoint from where the users will be able to perform semantic searches and exploit the data.

2. Terminology used

In the following sections, classes and properties could be defined as "mandatory", "recommended" and "optional" depending on its cardinality. These terms have the following meaning.

- **Mandatory class (1/1..*):** a receiver of data MUST be able to process information about instances of the class; a sender of data MUST provide information about instances of the class;

[1] EU Vocabularies Website: https://op.europa.eu/en/web/eu-vocabularies

[2] Wood, D., Lanthaler, M., & Cyganiak, R. (2014). RDF 1.1 Concepts and Abstract Syntax.

- **Recommended class (0..1/*):** a receiver of data MUST be able to process information about instances of the class; a sender of data MUST provide information about instances of the class, if it is available;
- **Optional class (0..1/*):** a receiver MUST be able to process information about instances of the class; a sender MAY provide the information but is not obliged to do so;
- **Mandatory property (1/1..*):** a receiver MUST be able to process the information for that property; a sender MUST provide the information for that property;
- **Recommended property (0..1/*):** a receiver MUST be able to process the information for that property; a sender SHOULD provide the information for that property if it is available;
- **Optional property (0..1/*):** a receiver MUST be able to process the information for that property; a sender MAY provide the information for that property but is not obliged to do so.

The meaning of the terms MUST, MUST NOT, SHOULD and MAY in this section and in the following sections are as defined in RFC 2119[3].

In the given context, the term "processing" means that receivers must accept incoming data and transparently provide these data to applications and services. It does neither imply nor prescribe what applications and services finally do with the data (parse, convert, store, make searchable, display to users, etc.).

3. Context of use

The use case that this Ontology intends to enable is publishing and exchange of all research data and related controlled vocabularies managed by the Publications Office of the European Union.

The basic use case involves the following actors:

- CORDIS team, which is in charge of the maintenance of EURIO and its related datasets;
- EU institutions and external stakeholders (practitioners, researches, public administrations, among others) who wish to (re-)use EURIO and its related datasets as Linked Open Data.

4. Used vocabularies

This Ontology reuses classes and properties from various existing specifications. Classes and properties specified in the next sections have been taken from the following namespaces.

Ontology	Prefix	URI
Simple Knowledge Organization System	skos	http://www.w3.org/2004/02/skos/core#
DCMI - Dublin Cores Metadata Terms Initiative	dcterms	http://purl.org/dc/terms/

[3] IETF. RFC 2119. Key words for use in RFCs to Indicate Requirement Levels. http://www.ietf.org/rfc/rfc2119.txt

OWL 2 Web Ontology Language	owl	http://www.w3.org/2002/07/owl#
Resource Description Framework	rdf	http://www.w3.org/1999/02/22-rdfsyntax-ns#
RDF Schema Vocabulary	rdfs	http://www.w3.org/2000/01/rdfschema#
XML Schema Definition	xsd	http://www.w3.org/2001/XMLSchema#
DINGO – Data INtegration for Grants Ontology	dg	https://w3id.org/dingo#
Schema.org	schema	http://schema.org/
FOAF – Friend Of a Friend	foaf	http://xmlns.com/foaf/0.1/
DCAT – Data Catalogue Vocabulary	dcat	http://www.w3.org/ns/dcat#
FRAPO – Funding, Research Administration and Projects Ontology	frapo	http://purl.org/cerif/frapo/
The Organization Ontology	org	http://www.w3.org/ns/org#
Project Ontology	oegp	webode://droz.dia.fi.upm.es/Project+Ontology#

Table 1*:* List of ontologies and their namespace definitions

There is also a list of controlled vocabularies used to restrict the value range on some properties. The values belong to the following namespaces.

Vocabulary	Prefix	URI
EuroSciVoc Taxonomy	esv	http://data.europa.eu/8mn/euroscivoc#

Table 2: List of controlled vocabularies and their namespace definitions

5. Graphical representation

The graphical representation of EURIO is provided in the form of an UML class diagram and is depicted in Figure 1. The boxes represent classes while the arrow connections represent properties establishing relations to other classes. The attributes inside boxes represent properties providing either literal data values or relation to other classes that omitted from the diagram.

The green boxes related with some classes represent taxonomies or controlled vocabularies that will specify through instances the type of an entity, knowledge area, business sector, and so on.

The cardinality specifications on the connector "_ .. _" arrows next to the relationships between classes represent constraints on how the property may be employed on the class instances and has a normative meaning. The first number means minimum cardinality constraint and the second means maximum cardinality constraint. The minimum cardinality constraint is zero "0 .. _ " for optional properties and one "1 .. _" for mandatory properties. The maximum cardinality constraint is usually unspecified " _ .. *" or limited to one "_ .. 1". If the cardinality is not specified then the implied meaning is exactly one "1".

The legend below the UML diagram indicates what are the classes that have been reused from other existing ontologies.

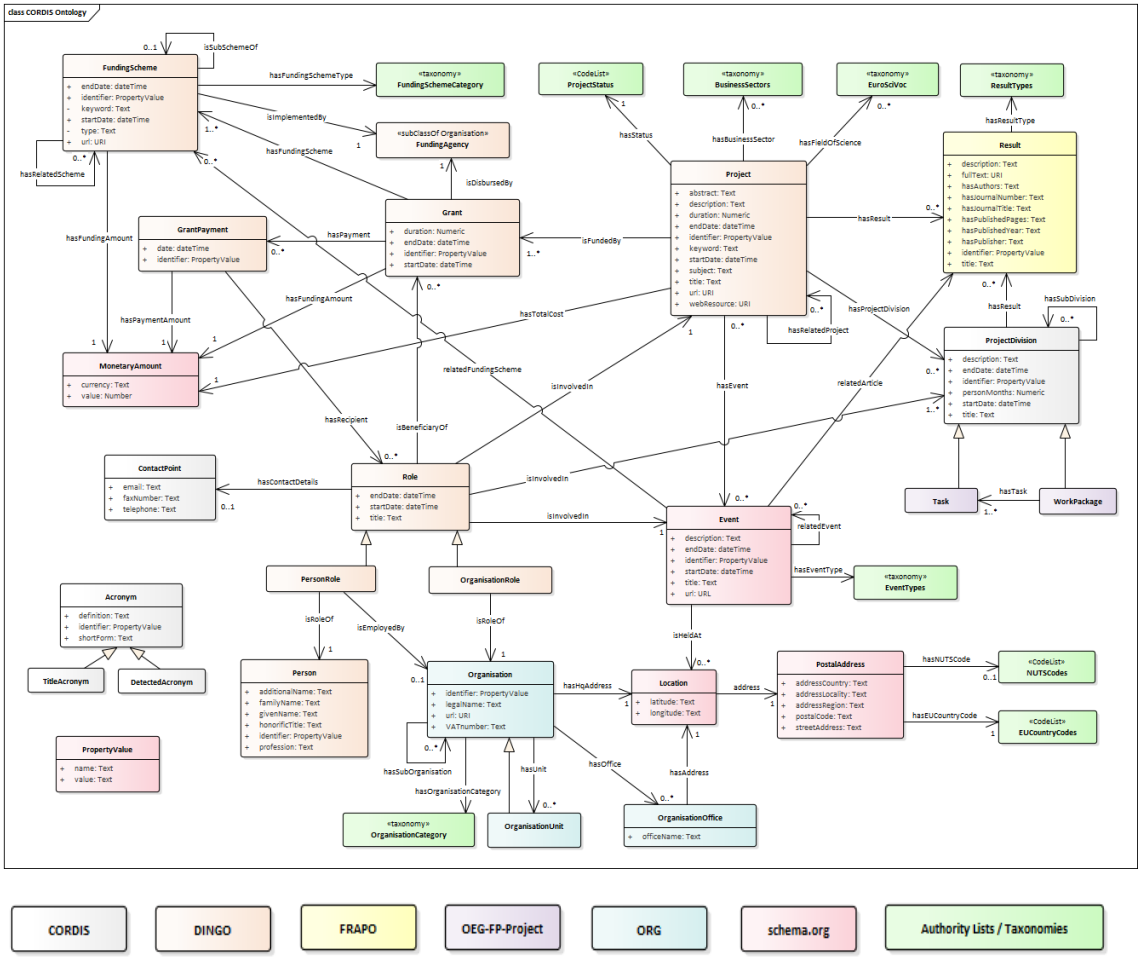

Figure 1: EURIO representation

6. EURIO

The following tables provide the definitions of the ontology classes and their properties, including their equivalent mappings (if any) with other ontologies.

Property value

This is a structured attribute that creates a property-value pair, e.g., representing a feature of a place. Its main use is for representing the different range of identifiers of classes, e.g., projects IDs. It is aligned with schema:PropertyValue.

Properties	Mappings	Type	Cardinality	Definition
name	-	xsd:string	1..1	The value of the acronym.
value	-	xsd:nonNegativeInteger	1..1	The definition of the acronym.
hasidentifier	schema:identifier	PropertyValue	1..*	A character string used to establish the identity of, and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme

Acronym

An acronym is an abbreviation formed from the initial letters of other words and pronounced as a word. It can refer to any type of entities in the CORDIS conceptual framework and may be stated explicitly or extracted automatically from text.

Properties	Mappings	Type	Cardinality	Definition
shortForm	-	xsd:string	1..*	The value of the acronym.
definition	-	xsd:string	0..*	The definition of the acronym.
hasAcronym	-	owl:Thing	0..*	Relates an entity (e.g. project, grant, event, etc.) with its acronym(s). This includes acronyms representing the entity's title as well as any additional acronyms extracted from other attributes of the entity. Inverse of *isAcronymOf*.
isAcronymOf	-	owl:Thing	0..*	Relates the acronym to the entity to which it belongs. Inverse of *hasAcronym*.

Title Acronym

A Title Acronym is an acronym explicitly defined as the title or name of another entity (e.g. Project, Organisation, Funding Scheme, etc.).

Properties
Inherited from Acronym (TitleAcronym is a sub-class of Acronym).

Detected Acronym

A Detected Acronym is any acronym that has been detected, automatically or manually, from the description of a class or from any other form of natural language text.

Properties
Inherited from Acronym (DetectedAcronym is a sub-class of Acronym).

Project

A project is a planned research work that has one or more objectives (divided or not into tasks) and is conducted by one or more organisations. In CORDIS, it represents a project funded by an EU programme. It is mapped to dg:Project

Properties	Mappings	Type	Cardinality	Definition
abstract	dg:abstract_or_summary_description	xsd:string	1..1	The general summary and ideas of a CORDIS project.
hasidentifier	schema:identifier	PropertyValue	1..*	A character string used to establish the identity of, and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme
duration	-	xsd:integer	1..1	The period of time an class (e.g. project, grant, event, etc.) has taken (use ISO 8601 duration format).
businessSector	-	skos:Concept	1..*	One or more values from the NACE taxonomy that represent the main business sectors of the project.
endDate	dg:end_time	xsd:dateTime	1..1	The date on which an entity (e.g. project, grant, event, etc.) is finalized or ceases to exist.
fieldOfScience	dg:field_of_work	skos:Concept	1..*	One or more values from the EuroSciVoc taxonomy that represent the main fields of science of the project.
keywords	-	xsd:string	1..*	Word or words that represent the main idea of the project.
startDate	dg:start_time	xsd:dateTime	1..1	The time an entity (e.g. project, grant, event, etc.) is initialized or starts to take effect.
status	-	skos:Concept	1..1	The current situation or progress of the project
subject	dg:main_topic	xsd:string	1..1	One word or phrase that describes the main subject, topic or area of the project.
description	schema:description	xsd:string	1..1	A brief summary of the entity (project, event, etc.).
title	dg:title	xsd:string	1..1	The entity's (e.g. project, work package, etc.) official name or title.
webLink	dg:official_website	xsd:anyURI	0..*	A URL that links to other internet resources outside CORDIS environment that are related with the project.

webResource	-	xsd:anyURI	0..*	A URL that links to resources other than webpages (e.g. document, image) that are related with the project.
hasTotalCost	dg:budget	MonetaryAmount	1	Relates the project with the amount of money expected, required, or given in payment for the project.
isFundedBy	dg:funded_by	Grant	1..*	Relates the project with the grant(s) that completely or partially fund the project. Inverse of *funds*.
hasEvent	-	Event	0..*	Relates the project with its event(s). Inverse of *isEventOf*.
hasResult	dg:product_or_material_produced	Result	0..*	Relates the project with its result(s). Inverse of *isResultOf*.
hasProjectDivision	-	ProjectDivision	0..*	Relates the project with its division(s). Inverse of *IsDivisionOf*.
hasInvolvedParty	dg:participant	Role	1..*	Relates the Project or Project Division to the roles that are involved in the project. Inverse of *isInvolvedIn*.
hasRelatedProject	-	Project	0..*	Represents a generic non parent-child relation between two CORDIS projects.

Grant

A grant is a disbursed fund paid to a recipient or beneficiary within the context of a funded research project. In the context of CORDIS, it represents a grant of a specific EU programme (e.g. an H2020 grant). It is mapped to dg:Grant.

Properties	Mappings	Type	Cardinality	Definition
hasidentifier	schema:identifier	PropertyValue	1..*	A character string used to establish the identity of, and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme
duration	-	xsd:integer	1..1	The period of time a class (e.g. project, grant, event, etc.) has taken (use ISO 8601 duration format).
startDate	dg:start_time	xsd:dateTime	1..1	The time an entity (e.g. project, grant, event, etc.) is initialized or starts to take effect.
endDate	dg:end_time	xsd:dateTime	1..1	The date on which an entity (e.g. project, grant, event, etc.) is finalized or ceases to exist.

isDisbursedBy	dg:disbursed_by	Funding Agency	1..1	Relates the grant to the funding agency that disburses the grant. Inverse of *disburses*.
hasFundingAmount	dg:economic_value	Monetary Amount	1..1	Relates the grant or funding scheme to the amount of money available for funding.
hasFundingScheme	dg:implementation_of	Funding Scheme	1..*	Relates the grant to the funding scheme of which the grant forms a part. Inverse of *hasGrant*.
hasPayment	dg:hasPart	Grant Payment	0..*	Relates the grant to its payment(s). Inverse of *isPaymentOf*.
funds	dg:finances	Project	1..*	Relates the grant with the project(s) that are completely or partially funded by it. Inverse of *isFundedBy*.
hasBeneficiary	dg:beneficiary	Role	0..*	Relates the grant to the organisation or person role(s) that are a beneficiary of the grant. Inverse of *isBeneficiaryOf*.

Grant Payment

A GrantPayment is a single payment to a recipient or beneficiary of a Grant within a research project. It is mapped to dg:GrantPayement

Properties	Mappings	Type	Cardinality	Definition
hasidentifier	schema:identifier	PropertyValue	1..*	A character string used to establish the identity of, and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme
Date	dg:point_in_time	xsd:dateTime	1..1	The date in which the grant payment took place.
hasPaymentAmount	dg:economic_value	Monetary Amount	1..1	Relates the payment to the amount of money that was given to the beneficiary of the payment.
isPaymentOf	dg:isPartOf	Grant	1..1	Relates a payment to the grant of which it is a part. Inverse of *hasPayment*.

hasRecipient	-	Role	0..*	Relates the grant payment to the organisation or person role(s) which have received the grant payment. Inverse property of *isRecipientOf*.

Funding Scheme

A FundingScheme is a Set of rules and conditions which forms the basis for public funding of European research. In CORDIS, they represent the different legal frameworks (e.g. FP6, FP7, H2020). They can also be used to represent the different calls inside each framework by using the recursive relation "partOfScheme". It is mapped to dg:FundingScheme.

Properties	Mappings	Type	Cardinality	Definition
hasidentifier	schema:identifier	PropertyValue	1..*	A character string used to establish the identity of, and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme
startDate	dg:start_time	xsd:dateTime	1..1	The time an entity (e.g. project, grant, event, etc.) is initialized or starts to take effect.
endDate	dg:end_time	xsd:dateTime	1..1	The date on which an entity (e.g. project, grant, event, etc.) is finalized or ceases to exist.
hasFundingSchemeCategory	-	skos:Concept	1..1	A list of terms specifying the different categories / types of funding schemes
webLink	dg:official_website	xsd:anyURI	0..*	A URL that links to other internet resources outside CORDIS environment that are related with the project.
hasFundingAmount	-	Monetary Amount	1..1	Relates the grant or funding scheme to the amount of money available for funding.
isImplementedBy	dg:is_implemented_by	Funding Agency	1..1	Relates the funding scheme with the funding agency that implements and administers the scheme. Inverse of *implements*.
isSubSchemeOf	dg:isPartOf	Funding Scheme	0..1	Relates the funding scheme to its parent scheme of which this funding scheme forms a part. Inverse of *hasSubScheme*.

hasSubScheme	dg:hasPart	Funding Scheme	0..1	Relates the funding scheme to its sub-schemes. Inverse of *isSubSchemeOf*.
hasGrant	dg:has_implementation	Funding Scheme	0..*	Relates a funding scheme with the grants that form part of it. Inverse of *hasFundingScheme.*

Funding Agency

A Funding Agency is an organisation that materially administrates funding schemes and disburses related grants. In the context of CORDIS, it represents the various European agencies responsible for funding research projects (e.g. ERC, EASME, REA, etc.). It is aligned with dg:FundingAgency.

Properties	Mappings	Type	Cardinality	Definition
Inherited properties from Organization (FundingAgency is a subclass of Organisation).				
implements	dg:implements	Funding Scheme	0..*	Relates a funding agency to the funding schemes that it implements and administers. Inverse of *isImplementedBy*.
disburses	dg:disburses	Grant	1..1	Relates the funding agency to the grant that it disburses. Inverse of *isDisbursedBy*.

Project Division

A project division is an abstract entity that represents a part of the work and activities that need to be carried out within a project. This entity can be specialized through different sub-entities to represent the different types of project divisions.

Class	Mappings	Type	Cardinality	Definition
hasidentifier	schema:identifier	PropertyValue	1..*	A character string used to establish the identity of, and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme
description	schema:description	xsd:string	1..1	A brief summary of the entity (project, event, etc.).
title	oegp:name	xsd:string	1..1	The entity's (e.g. project, work package, etc.) official name or title.
startDate	oegp:start_date	xsd:dateTime	1..1	The time an entity (e.g. project, grant, event, etc.) is initialized or starts to take effect.

endDate	oegp:end_date	xsd:dateTime	1..1	The date on which an entity (e.g. project, grant, event, etc.) is finalized or ceases to exist.
person Months	oegp:person_months	xsd:nonNegativeInteger	0..1	The effort needed to carry out work contained in the project division, estimated in person-months.
hasResult	-	Result	0..*	Relates the project division to the result(s) stemming from the work contained in the division. Inverse of *isResultOf*.
isDivisionOf	-	ProjectDivision	0..*	Relates the project division to the project of which it is part. Inverse of *hasProjectDivision*.
hasInvolved Party	dg:participant	Role	1..*	Relates the Project or Project Division to the roles that are involved in the project. Inverse of *isInvolvedIn*.

Work Package

A Work Package is a major subdivision of a project which leads to the completion of one of the goals, objectives or major deliverables within the project. Different work packages can proceed in parallel within a project. It is aligned with oegp:WorkPackage

Properties	Mappings	Type	Cardinality	Definition
Inherited properties from ProjectDivision (WorkPackage is a sub-class of ProjectDivision).				
hasTask	oegp:isMadeUpOf	Task	1..*	Relates the work package to the task(s) that make up the package. Inverse of *isTaskOf*.

Task

A task is a sub-division of a Work Package. Different tasks can proceed in parallel, within a work package, and cover one or more reporting periods of the project. A task must start and end with the time range of its parent Work Package. It is aligned with oegp:Task.

Properties	Mappings	Type	Cardinality	Definition
Inherited properties from ProjectDivision (WorkPackage is a sub-class of ProjectDivision).				
isTaskOf	oegp:isMadeUpOf	Task	1..*	Relates the work package to the task(s) that make up the package. Inverse of *isTaskOf*.

Result

A Result is any tangible or intangible output of the project (such as data, knowledge and information, whatever their form or nature, whether or not they can be protected), which are generated in the project. It is aligned with frapo:Output.

Properties	Mappings	Type	Cardinality	Definition
description	schema:description	xsd:string	1..1	A brief summary of the entity (project, event, etc.).
hasidentifier	schema:identifier	Property Value	1..*	A character string used to establish the identity of, and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme
title	dg:title	xsd:string	1..1	The entity's (e.g. project, work package, etc.) official name or title.
hasPublishedYear		xsd:string	1..*	Represents the year when this Result was published.
hasAuthor		xsd:string	1..*	Represents the list of authors that have contributed to the creation of the Result.
hasJournalNumber		xsd:string	0..*	Represents the journal number of the Result.
hasJournalTitle		xsd:string	0..*	Represents the title of the journal where the information about the result can be located.
hasPublishedPages		xsd:string	0..*	Represents the number of pages that this Result is published under.
hasPublisher		xsd:string	1..*	Represents the publishing body of this Result.
fullText		xsd:string	0..*	Represents the location of the full text of the Result online.
isResultOf		Project Division ; Project	0..*	Relates the result to the project or project division of which the result is an outcome. Inverse of *hasResult*.

Event

An Event is an happening at a certain time and location, such as a meeting, lecture, or conference. In CORDIS, it represents the events are organized as part of or are related to research projects. It is aligned with schema:Event.

Properties	Mappings	Type	Cardinality	Definition
description	schema:description	xsd:string	1..1	A brief summary of the entity (project, event, etc.).

hasidentifier	schema:identifier	Property Value	1..*	A character string used to establish the identity of, and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme
hasEventType		skos:Concept	1..1	A controlled list of types of events
startDate	dg:start_time	xsd:dateTime	1..1	The time an entity (e.g. project, grant, event, etc.) is initialized or starts to take effect.
endDate	dg:end_time	xsd:dateTime	1..1	The date on which an entity (e.g. project, grant, event, etc.) is finalized or ceases to exist.
title	dg:title	xsd:string	1..1	The entity's (e.g. project, work package, etc.) official name or title.
webLink	dg:official_website	xsd:anyURI	0..*	A URL that links to other internet resources outside CORDIS environment that are related with the project.
isHeldAt	schema:location	Location	0..*	The location where the event is happening.
isEventOf	-	Project	0..1	Relates the event to the project of which it is part. Inverse of *hasEvent*.

Role

A Role is the function assumed by or ascribed to an entity (typically a person or organisation) in a particular project. It can be used to precise or represent additional information about a relationship or property such as participation or membership. In CORDIS, it represents the participation of a person or organisation in a research project. It is aligned with dg:Role.

Properties	Mappings	Type	Cardinality	Definition
title	dg:title	xsd:string	1..1	The entity's (e.g. project, work package, etc.) official name or title.
startDate	dg:start_time	xsd:dateTime	1..1	The time an entity (e.g. project, grant, event, etc.) is initialized or starts to take effect.
endDate	dg:end_time	xsd:dateTime	1..1	The date on which an entity (e.g. project, grant, event, etc.) is finalized or ceases to exist.
isBeneficiaryOf	dg:beneficiary_of	Grant	0..*	Relates the role to the grant(s) of which the role is a beneficiary. Inverse of *hasBeneficiary.*
hasContactDetails	-	Contact Details	0..1	Relates the role to the contact details of the entity taking the role (person or organisation) which are valid during the project of which the role forms a part.

isInvolvedIn	dg:participant_of	Project Division ; Project	1..1	Relates the role to the Project or Project Division in which it is involved. Inverse of *hasInvolvedParty*.
isRecipientOf	-	Grant Payment	1..*	Relates the organisation or person role with the grant payment(s) which they have received. Inverse property of hasRecipient.

Person Role

A Person Role is a Function assumed by or ascribed to a person that takes part in a particular project. In CORDIS, it represents roles such as contact person, PI, etc.

Properties	Mappings	Type	Cardinality	Definition
Inherited properties from Role (PersonRole is a sub-class of Role).				
isEmployedBy	dg:employer	Organisation	0..1	Relates the person role to the organisation in which the person is employed during the project. Inverse of *employs*.
isRoleOf	dg:entity_taking _role	Person	1..1	Relates the person role to the person which assumes the role. Inverse of *hasPersonRole*.

Organisation Role

An Organisation Role is a function assumed by or ascribed to an organisation that takes part in a particular project. In CORDIS, it represents roles such as coordinator, partner, consortium member, etc.

Properties	Mappings	Type	Cardinality	Definition
Inherited properties from Role (PersonRole is a sub-class of Role).				
isRoleOf	dg:entity_taking _role	Person	1..1	Relates the person role to the person which assumes the role. Inverse of *hasPersonRole*.

Person

A Person represents the different people assuming roles in research projects (e.g. PhD students, PIs, administrative contacts, etc.). It is aligned with dg:Person.

Properties	Mappings	Type	Cardinality	Definition
givenName	schema:givenName	xsd:string	1..1	The given name (first name) of the person.
honorificTitle	schema:honorificPrefix	xsd:string	1..1	An honorific prefix preceding a Person's name such as Dr, Mrs, Mr.
familyName	schema:familyName	xsd:string	1..1	The family name (last name) of the person.

profession	schema:hasOccupation	xsd:string	1..*	The Person's occupation. In CORDIS, the values of this property may come from the NACE taxonomy.
additionalName	schema:additionalName	xsd:string	1..*	An additional name for a Person, can be used for a middle name.
hasidentifier	schema:identifier	Property Value	1..*	A character string used to establish the identity of, and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme
hasPersonRole	schema:takes_role	Role	1..*	Relates the person to the roles that the person takes. Inverse of *isRoleOf*.

Organisation

An Organisation is a group of people, with the same purpose, organized together into a community or other social, commercial or political structure that participate in CORDIS projects. It is aligned with org:Organization.

Properties	Mappings	Type	Cardinality	Definition
legalName	schema:legalName	xsd:string	1..1	The official name of the organization, e.g. the registered company name.
hasidentifier	schema:identifier	PropertyValue	1..*	A character string used to establish the identity of, and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme
hasOrganisationCategory		skos:Concept	1..1	A list of terms specifying the different types of organisations
VATNumber	schema:vatID	xsd:string	1..1	The unique number that identifies a taxable organisation or non-taxable legal entity that is registered for VAT.
webLink	dg:official_website	xsd:anyURI	0..*	A URL that links to other internet resources outside CORDIS environment that are related with the project.
hasOffice	org:hasSite	Organisation Office	0..*	Relates the organisation with a physical office which belongs to the organisation. Inverse of *isOfficeOf*.
hasUnit	org:hasUnit	Organisation Unit	0..*	Relates the organisation with a unit or department that forms part of the organisation. Inverse of *isUnitOf*.

hasHqAddress	org:hasPrimarySite	Location	1..1	Relates the organisation with the official, registered address of the headquarters or primary offices of the organisation.
hasSubOrganisation	org:hasSubOrganization	Organisation	0..*	Relates the organisation with another organisation which is a sub-part or child of it. Inverse of *isSubOrganisationOf*.
isSubOrganisationOf	org:subOrganizationOf	Organisation	0..*	Relates the organisation with another organisation of which it is a sub-part. Inverse of *hasSubOrganisation*.
hasOrganisationRole	dg:takes_role	OrganisationRole	0..*	Relates the organisation to the roles that the organisation takes. Inverse of *isRoleOf*.
employs		PersonRole	0..*	Relates the organisation to the role representing a person that the organisation employs. Inverse of *isEmployedBy*.

Organisation Unit

An Organisational Unit is a department or unit which is part of some larger Organization and only has full recognition within the context of that Organization. In CORDIS, it represents for instance the different departments of a university that participate in a particular project. It is aligned with org:OrganizationalUnit.

Properties	Mappings	Type	Cardinality	Definition
Inherited properties from Organisation (OrganisationUnit is a subclass of Organisation).				
isUnitOf	Org:unitOf	Person	1..*	Relates the unit to the organisation of which the unit forms a part. Inverse of *hasUnit*.

Organisation Office

An Organisation Office is an office or other premise at which the organization is located. Many organizations are spread across multiple sites and many sites will host multiple locations. In CORDIS, it represents the different offices of an organisation or a unit that participates in a particular project. It is aligned with org:Site.

Properties	Mappings	Type	Cardinality	Definition
officeName	-	xsd:string	1..1	The name or title of the office.
hasAddress	org:siteAddress	Location	1..1	Relates the office to its office registered address.
isOfficeOf	org:siteOf	Organisation	1..1	Relates the office to the organisation to which it belongs. Inverse of *hasOffice*.

Location

A Location is the coordinates of an entity that has a fixed physical location. In CORDIS, it is used to represent the addresses of organisations that participate in research projects. It is aligned with schema:Geocoordinates.

Properties	Mappings	Type	Cardinality	Definition
Latitude	schema:latitude	xsd:string	1..1	The latitude of a location. For example, 37.42242 (WGS 84).
Longitude	schema:longitude	xsd:string	1..1	The longitude of a location. For example, 122.08585 (WGS 84).
Address	schema:address	xsd:string	1..1	Physical address of the item.

Postal Address

A Postal Address is the postal address of an entity that has a fixed physical location. In CORDIS, it is used to represent the addresses of organisations that participate in research projects. It is aligned with schema:PostalAddress

Properties	Mappings	Type	Cardinality	Definition
addressCountry	schema: addressCountry	xsd:string	1..1	The country of the place or address. You can also provide the two-letter ISO 3166-1 alpha-2 country code.
addressLocality	schema: addressLocality	xsd:string	1..1	The locality, town or city of the place or address.
addressRegion	schema: addressRegion	xsd:string	1..1	The administrative area, region or state of the place or address.
postalCode	schema: postalCode	xsd:string	1..1	The postal code of the place or address.
streetAddress	streetAddress	xsd:string	1..1	The street name and number of the place or address.
euCountryCode		skos:Concept	1..1	The country code of the place or address according to the EU country codes.
NUTSCode		skos:Concept	1..1	A PropertyValue pair representing the NUTS-2 or NUTS-3 code of the place or address.

Contact Point

A Contact Point represents a contact point of an organisation or a person. It is aligned with schema:ContactPoint.

Properties	Mappings	Type	Cardinality	Definition
email	schema: email	xsd:anyURI	0..*	Email address.
faxNumber	schema: faxNumber	xsd:string	0..*	The fax number.

telephone	schema: telephone	xsd:string	0..*	The telephone number.

7. Concept evolution and versioning

The publication of EURIO will take place at most every 6 months in the case there are modifications to be made. This will depend on whether there have been enough changes warranting new releases of the ontology.

The Ontology will be reviewed approximately every 6 months. During this revision process, the following tasks are carried out:

1. **CORDIS data model update:** this task is based on multiple factors:
    a. The new requirements identified from curation of CORDIS data;
    b. The change request by its stakeholders;
    c. The evolution and alignment with the reused ontologies such as DINGO, Schema.org, ORG., etc. If there is a modification in these vocabularies that have an impact on EURIO, this will be updated as well.
2. Update the OWL file (in Turtle).

Once the data model and the OWL file are updated according to the identified needs, CORDIS produces a new release of the ontology to be published.