International Standard

ISO/IEC 11179-34

First edition
2024-05

Information technology — Metadata registries (MDR) —

Part 34: Metamodel for computable data registration

Technologies de l'information — Registres de métadonnées (RM) —

Partie 34: Métamodèle pour l'enregistrement des données calculables

Reference number
ISO/IEC 11179-34:2024(en)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

This document is part of the fourth modularization of ISO/IEC 11179. This document brings into ISO/IEC 11179 the ability to register information about computable data.

A list of all parts in the ISO/IEC 11179 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

Significant scientific discoveries are increasingly achieved through complex and distributed computations and data analyses. These computations and analyses always involve processing files through a series of steps and transformations, usually called a pipeline or a workflow.

Data pipelines typically use multiple pieces of software, each of which typically has multiple versions available, multiple input parameters, multiple outputs, and possibly platform-specific configurations. As with experimental parameters in a laboratory protocol, small changes in computational parameters can have a large impact on the scientific validity of the results.

To reproduce and verify scientific discoveries, details of pipelines need to be documented and shared, including the protocol, procedures, or metadata associated with them. The more analysis steps and the more complicated a pipeline, the greater the need for a standardized mechanism of communication. A detailed communication helps ensure responsibility, reproducibility, and the ability to verify protocol, track provenance information, and promote interoperability.

This document is inspired by the IEEE 2791-2020[1], which aims to improve communication of bioinformatics protocols and data to facilitate bioinformatics workflow related exchange and communication between regulatory agencies, pharmaceutical companies, bioinformatics platform providers and researchers. Although IEEE 2791-2020 has a bioinformatics background and application areas, the concepts and methods it expressed are applicable to a broader field of scientific research. A mapping table is included in Annex A, showing the relationship between domains and fields in IEEE 2791-2020 and classes and attributes in this document.

ISO/IEC 11179-3 specifies the structure of a Metadata Registry (MDR) and provides a metamodel for registry common facilities. That metamodel is intended to be extended by other parts of ISO/IEC 11179 for specific purposes.

This document provides a specification of the extensions to the registry metamodel specified in ISO/IEC 11179-3 to enable the registration of metadata about computable data. Registration of metadata about computable data are like a manifest describing all details related to input files, output files, and the pipeline used to process these files. The intent is to facilitate efficient communication and interoperability among different platforms, industries, scientists, and regulators and to improve reproducibility and replicability.

In Clauses 6 and 7, this document uses **bold** font to highlight terms which represent metadata objects specified by the metamodel.

EXAMPLE **Computable_Data** (see 7.2.2.1) is a class each instance of which models computable data.

Information technology — Metadata registries (MDR) —

Part 34:
Metamodel for computable data registration

1 Scope

This document provides a specification for an extension to a metadata registry (MDR), as specified in ISO/IEC 11179-3, in which metadata that describe computable data can be registered.

The specification in this document, together with the relevant clauses of the specification in ISO/IEC 11179-3, provides the ability to record metadata about computable data.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 11179-3, *Information technology — Metadata registries (MDR) — Part 3: Metamodel for registry common facilities*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 11179-3 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
computable data
data that are computed for the purpose of making the data reproducible

3.2
pipeline
full set of logically connected steps needed to transform input data into a result, for which one or more outputs of one data transformation step can be the input of one or more subsequent steps

3.3
computation step
individual tool (or a well defined and reusable script) used in a *pipeline* (3.2) that can be executed sequentially or in parallel with other computation steps

3.4
computation execution environment
information of the environment needed to execute a *computation step* (3.3) to the extent to which consistent results are obtained when produced repeatedly, including platform, deployment, software configuration and running applications, etc.

4 Abbreviated terms

JSON JavaScript Object Notation

ORCID Open Researcher and Contributor ID (see Reference [2])

PAV Provenance, Authoring and Versioning (see Reference [3])

UML Unified Modeling Language (see References [4] and [5])

URI Uniform Resource Identifier

URL Uniform Resource Locator

5 Conformance

5.1 Overview of conformance

Conformance rules for a Metadata Registry are specified in ISO/IEC 11179-3:2023, Clause 4. The clause "Degree of Conformance" is repeated here for convenience. The subsequent subclauses extend the rules from ISO/IEC 11179-3:2023.

5.2 Degree of conformance

5.2.1 General

The distinction between "strictly conforming" and "conforming" implementations is necessary to address the simultaneous needs for interoperability and extensions. This document describes specifications that promote interoperability. Extensions are motivated by needs of users, vendors, institutions, and industries, and:

a) are not directly specified by this document;

b) are specified and agreed to outside this document;

c) may serve as trial usage for future editions of this document.

A strictly conforming implementation can be limited in usefulness but is maximally interoperable with respect to this document. A conforming implementation can be more useful but can be less interoperable with respect to this document.

5.2.2 Strictly conforming implementations

A strictly conforming implementation:

a) shall support all mandatory, optional and conditional classes, attributes, datatypes and associations;

b) shall not use, test, access or probe for any extension features nor extensions to classes, attributes, datatypes, associations or any combination thereof;

c) shall not recognize, nor act on, nor allow the production of classes, attributes, datatypes, associations or any combination thereof that are dependent on any unspecified, undefined or implementation-defined behaviour.

NOTE The use of extensions to the metamodel can cause undefined behaviour.

5.2.3 Conforming implementations

A conforming implementation:

a) shall support all mandatory, optional and conditional classes, attributes, datatypes and associations;

b) as permitted by the implementation, may use, test, access or probe for extension features or extensions to classes, attributes, datatypes, associations or any combination thereof;

c) may recognize, act on or allow the production of classes, attributes, datatypes, associations or any combination thereof that are dependent on implementation-defined behaviour.

NOTE 1 All strictly conforming implementations are also conforming implementations.

NOTE 2 The use of extensions to the metamodel can cause undefined behaviour.

5.3 Conformance by feature

Conformance claims may be made to Clause 7 of this document, or to specific features within that clause. Clause 7 is also dependent upon one or more clauses of ISO/IEC 11179-3, so conformance to all or part of Clause 7 shall be understood to imply conformance also to relevant provisions specified in one or more of the clauses in ISO/IEC 11179-3.

A conformance statement shall specify exactly the features supported and not supported.

5.4 Registry conformance

5.4.1 Standard registry profiles

This document specifies the following standard profiles in addition to those specified in ISO/IEC 11179-3:2023, 4.4.2.

— **Computable data Registry**: Implements Clause 7 of this document, in addition to all provisions of the Basic registry profile of ISO/IEC 11179-3:2023, 4.4.2;

— **Computable data Registry with mapping**: Implements Clause 7 of this document, in addition to all provisions of the Basic registry with mapping profile of ISO/IEC 11179-3:2023, 4.4.2.

5.4.2 Conformance labels

Conformance to the profiles specified in 5.4.1 may be claimed using the following labels, respectively:

— ISO/IEC 11179-34:2024 Computable data Registry;

— ISO/IEC 11179-34:2024 Computable data Registry with mapping.

5.5 Implementation conformance statement (ICS)

An implementation claiming conformance to this document shall include an implementation conformance statement stating:

a) whether it conforms or strictly conforms;

b) which clauses are or are not supported;

c) what extensions, if any, are supported or used.

A standard profile may be referenced, if applicable.

EXAMPLE Product Z strictly conforms to ISO/IEC 11179-34:2024 Computable data Registry.

5.6 Obligation

Properties and relationships specified in this document are one of: Mandatory, Conditional or Optional. The obligation is not explicitly stated but is to be inferred from the multiplicity of the property or relationship, and the presence or absence of a condition.

For the purpose of conformance:

a) mandatory properties and relationships shall exist and shall conform to the provisions of this document;

b) anything specified as Conditional within this document shall be treated as Mandatory if the associated condition is satisfied and shall otherwise be not present;

c) optional properties and relationships are not required to exist, but if they do exist, they shall conform to the provisions of this document.

Such obligation is enforced if and only if the Registration Status of the associated registry items is Recorded or higher (see ISO/IEC 11179-3:2023, 9.4.4.3 and ISO/IEC 11179-6:2023, 4.3.4).

6 Relationship to ISO/IEC 11179-3

6.1 Metamodel for a metadata registry

A metamodel is a model that describes other models. A metamodel provides a mechanism for understanding the precise structure and components of the specified models, which are needed for the successful sharing of the models by users, software facilities or both.

ISO/IEC 11179-3 uses a metamodel to describe the information model of a metadata registry. The registry in turn will be used to describe and model other data, for example about enterprise, public administration or business applications. The registry metamodel is specified as a conceptual data model, i.e. one that describes how relevant information is structured in the natural world. In other words, it is how the human mind is accustomed to thinking of the information.

6.2 Specification of the metamodel

The conventions used in specifying the metamodel are described in ISO/IEC 11179-3:2023, 5.3. Many of the classes specified in this document inherit from *Item*, which is specified in ISO/IEC 11179-3:2023, 6.4.2.1. As *Items*, instances of these classes may be identified, registered, administered, named, defined and classified.

6.3 Use of UML class diagrams and textual description

This document uses both text and UML class diagrams to describe the metamodel. Both are normative and are intended to be complementary. However, if a conflict exists between what is specified in the UML class diagrams and what is specified in text, the text takes precedence until a correction is made to make them consistent. Further, if a conflict exists between a formal definition and other normative text, the formal definition takes precedence until a correction is made to make them consistent.

A consolidated UML class hierarchy is included as Annex B.

While the model diagrams are presented in UML class diagram notation, this document does not assume nor endorse any specific system environment, database management system, database design paradigm, system development methodology, data definition language, command language, system interface, user interface, computing platform, or any technology required for implementation.

6.4 Package dependencies

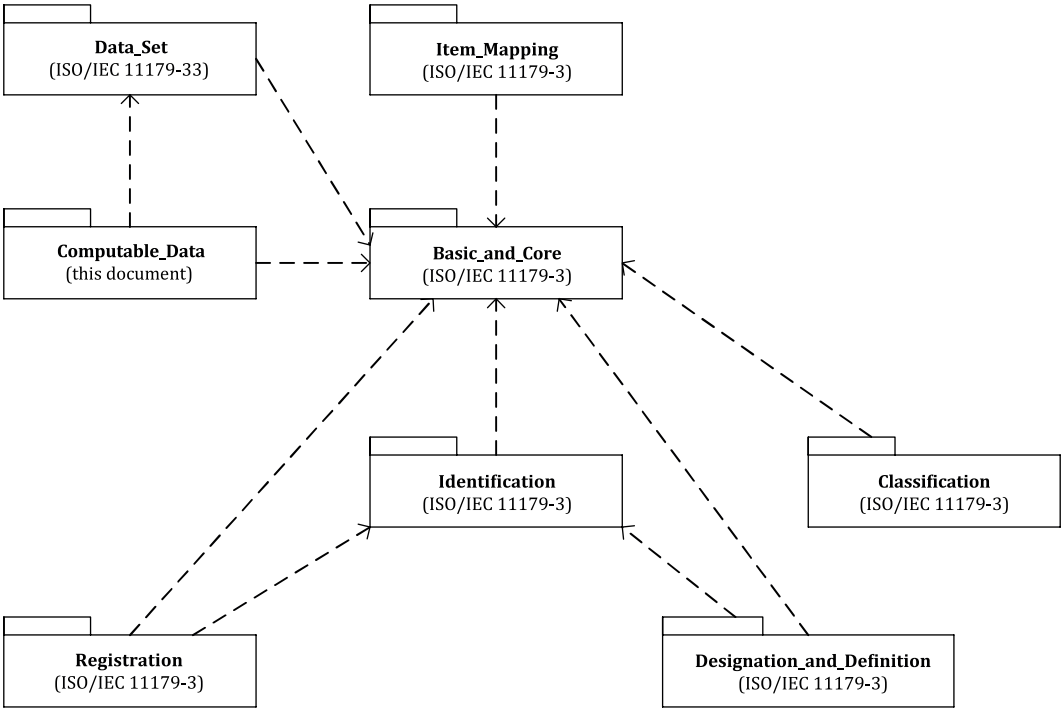

Figure 1 — Package dependencies

Figure 1 illustrates the dependencies among the packages. The Computable_Data package is specified in this document. All the other packages are specified in ISO/IEC 11179-3 and ISO/IEC 11179-33.

The lines in Figure 1 illustrate dependencies in the direction of the arrow. In order to implement a package that has dependencies, the packages on which it is dependent shall also be implemented. The dependencies are of three types:

a) subclassing from classes in another package, e.g. **Computable_Data** (see 7.2.2.1) in the Computable_Data package is subclassed from the **Item** class in the Basic and Core package (ISO/IEC 11179-3:2023, 6.4.2.1);

b) relationship between classes, e.g. **Registered_Item** in the Registration package (ISO/IEC 11179-3:2023, 9.4.1) has a relationship with **Reference_Document** in the Basic and Core package (ISO/IEC 11179-3:2023, 6.3.8);

c) some attributes use a predefined datatype or a class from another package as a datatype, e.g. the **supporting_document** attribute of the **Supporting_Document** (see 7.2.2.3) class in the Computable_Data package (see Clause 7) uses the **Reference_Document** class of the Basic and Core package (see ISO/IEC 11179-3:2023, 6.3.8) as a datatype.

Conformance options are specified in Clause 5 and standard conformance profiles in 5.4.

7 Computable_Data package

7.1 Overview of the Computable_Data package

The **Computable_Data** package consists of a single metamodel region, the Computable_Data metamodel region.

7.2 Computable_Data metamodel region

7.2.1 Overview of the Computable_Data metamodel region

Figure 2 shows the framework of the metamodel for the registration of metadata about computable data, Figure 3 and Figure 4 show the detailed information of this metamodel.

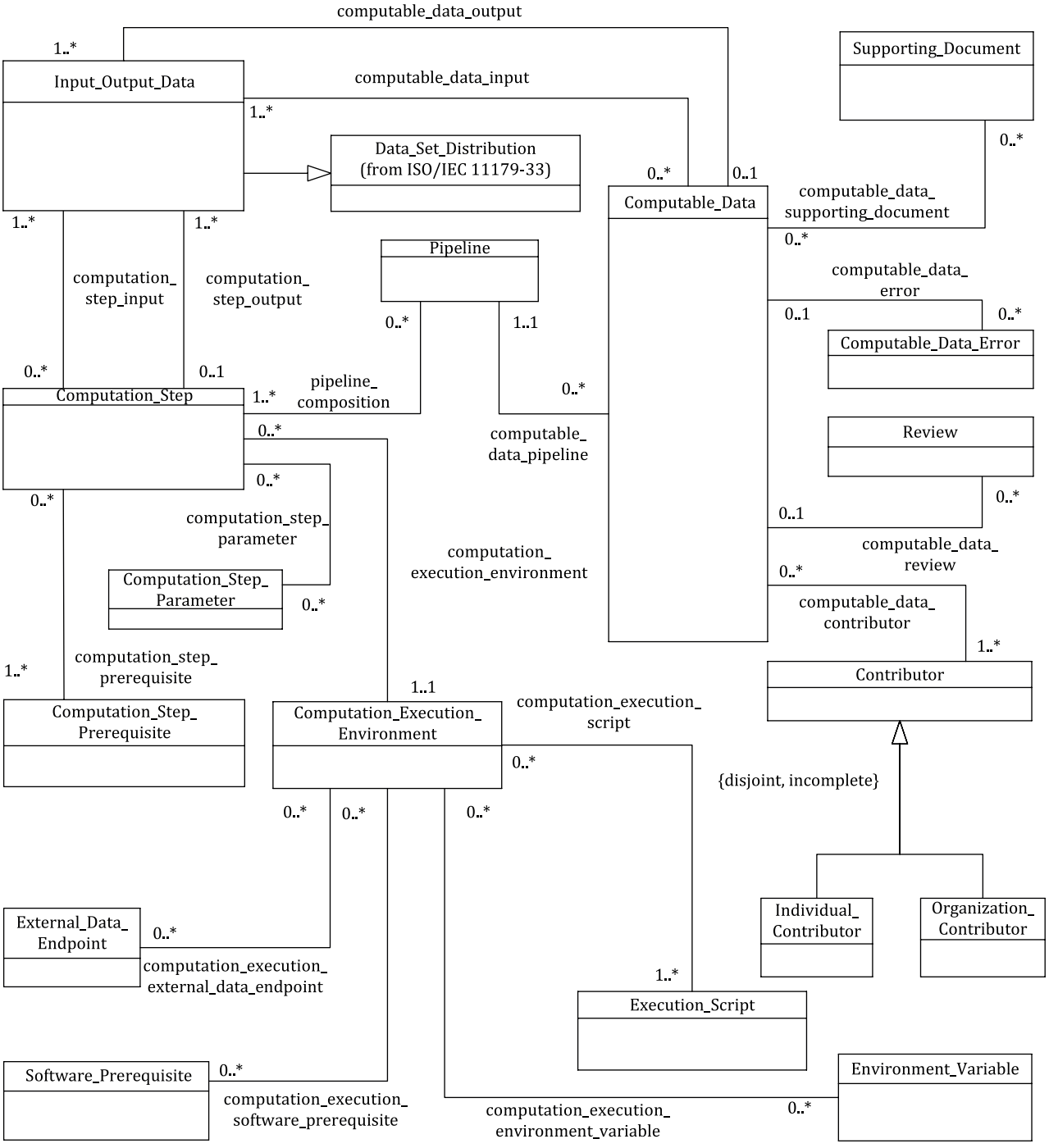

Figure 2 — Framework of computable data metamodel region

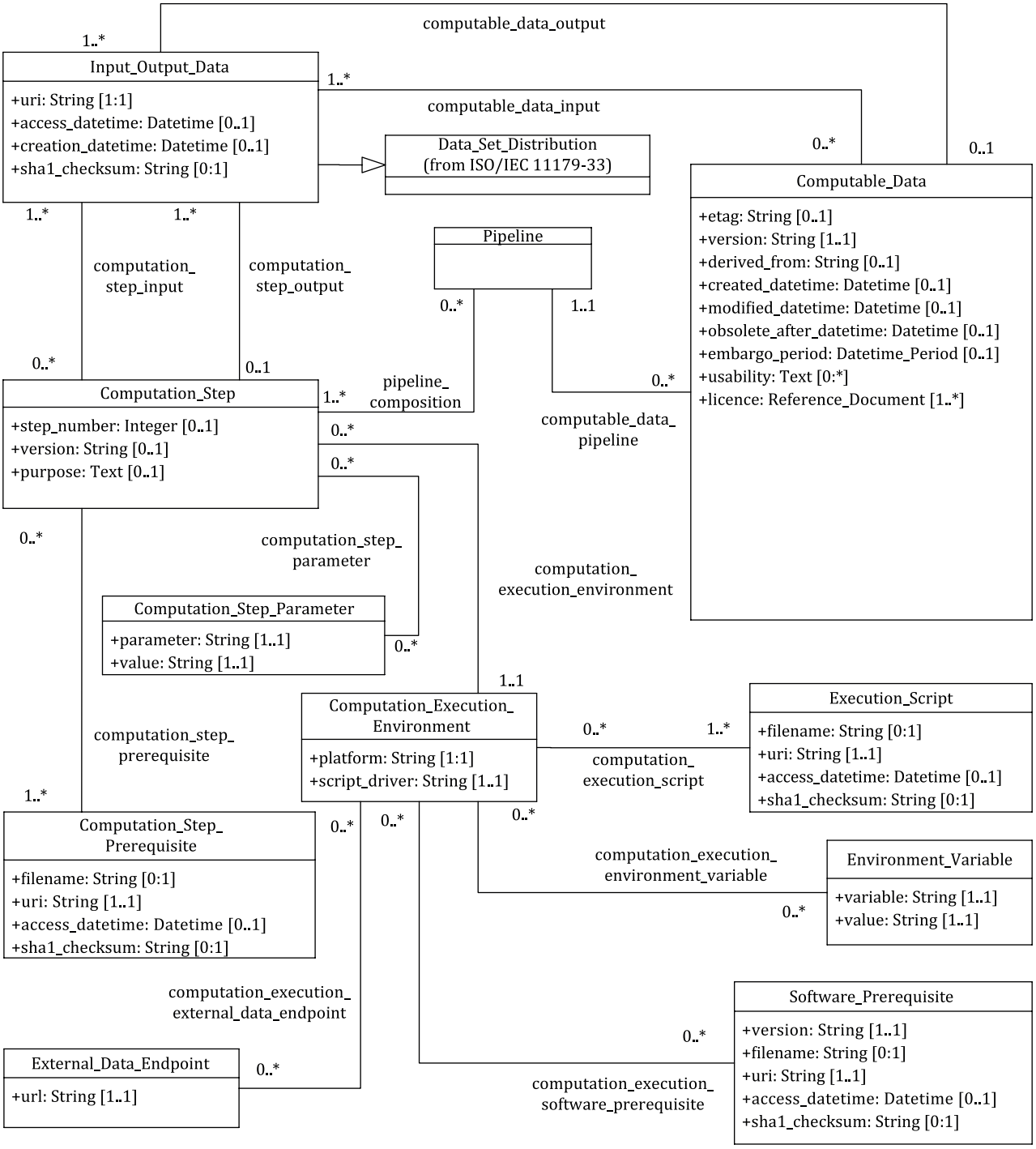

Figure 3 — Computable data metamodel region (1 of 2)

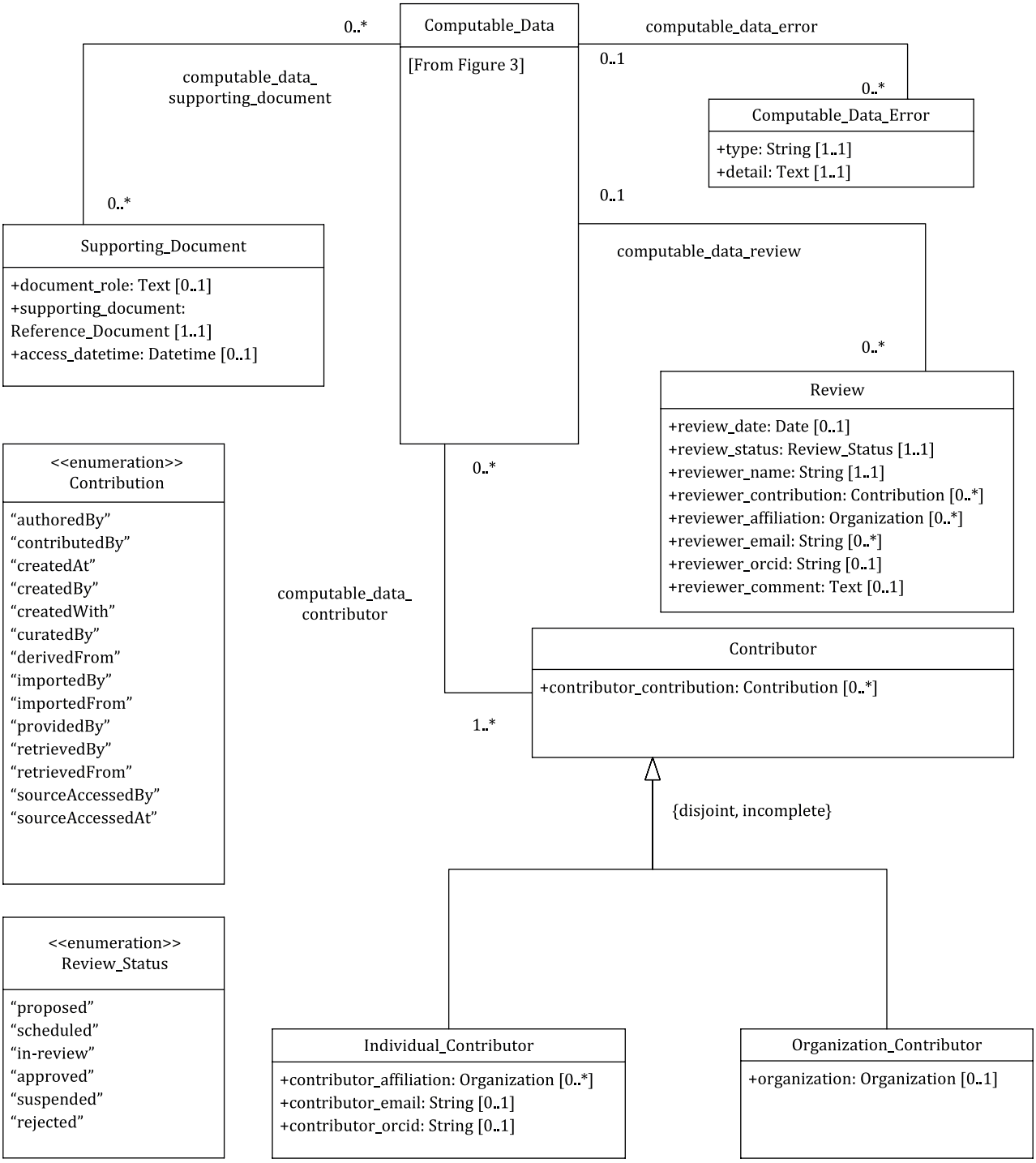

Figure 4 — Computable data metamodel region (2 of 2)

Examples of the registration of computable data using this metamodel are included in Annex C.

7.2.2 Classes in the Computable_Data metamodel region

7.2.2.1 Computable_Data class

7.2.2.1.1 Direct superclass

Computable_Data is a subclass of **Item** (see ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.1.2 Description of Computable_Data

Computable_Data is a class each instance of which models computable data, which are data that are computed for the purpose of making the data reproducible.

7.2.2.1.3 Associations of Computable_Data

As a subclass of **Item**, **Computable_Data** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2). **Computable_Data** has the following additional associations:

— **computable_data_pipeline** (see 7.2.3.1);

— **computable_data_supporting_document** (see7.2.3.2);

— **computable_data_error** (see 7.2.3.3).

— **computable_data_contributor** (see 7.2.3.4);

— **computable_data_review** (see 7.2.3.5);

— **computable_data_input** (see 7.2.3.6);

— **computable_data_output** (see 7.2.3.7).

7.2.2.1.4 Attributes of Computable_Data

The attributes of the **Computable_Data** class are specified in Table 1.

Table 1 — Attributes of the Computable_Data class

Attribute name	Multiplicity	Datatype	Description
etag	0..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: a string-type, read-only value, protecting the computable data from internal or external alterations without proper validation. NOTE: See RFC7232[6] section 2.3 for full description.
version	1..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: a version number which identifies the state of development of the computable data. In computable data versioning, a change affecting the outcome of the computation should be deposited as a new computable data, not as a new version. Changes that cannot affect the results of the computation can be incorporated into a new version of the existing computable data.
derived_from	0..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: the parent computable data that the computable data are derived from. NOTE: If the computable data are derived from another, this attribute will specify the parent computable data, in the form of its identifier.
created_datetime	0..1	**Datetime** (ISO/IEC 11179-3:2023, 6.2.3)	Definition: date and time of the initial creation of the computable data.
modified_datetime	0..1	**Datetime** (ISO/IEC 11179-3:2023, 6.2.3)	Definition: date and time the computable data was last modified.
obsolete_after_datetime	0..1	**Datetime** (ISO/IEC 11179-3:2023, 6.2.3)	Definition: expiration date and time of the computable data.

Table 1 *(continued)*

Attribute name	Multiplicity	Datatype	Description
embargo_period	0..1	**Datetime_Period** (ISO/IEC 11179-3:2023, 6.3.10)	Definition: period during which the computable data shall not be made public.
usability	0..*	**Text** (ISO/IEC 11179-3:2023, 6.2.12)	Definition: description of the scientific use case and function of the computable data.
licence	1.. *	**Reference_Document** (ISO/IEC 11179-3:2023, 6.3.8)	Definition: a document that provides licence information related to the computable data. NOTE: A public copyright licence (i.e. Creative Commons Attribution 4.0 International[7])is used when an author wants to give other people the right to share, use, and build upon a work that they (the author) have created.

7.2.2.1.5 Constraint on Computable_Data

Each instance of the **Computable_Data** class shall only exist if at least one instance of the **item_designation** association of the parent **Item** class shall be instantiated.

7.2.2.2 Pipeline class

7.2.2.2.1 Direct superclass

Pipeline is a subclass of **Item** (see ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.2.2 Description of Pipeline

Pipeline is a class each instance of which models a pipeline, a full set of logically connected steps needed to transform input data into a result, for which the output of one data transformation step can be the input of one or more subsequent steps.

7.2.2.2.3 Associations of Pipeline

As a subclass of **Item**, **Pipeline** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2). **Pipeline** has the following additional associations:

— **computable_data_pipeline** (see 7.2.3.1);

— **pipeline_compostion** (see 7.2.3.8).

7.2.2.2.4 Attributes of Pipeline

None.

7.2.2.3 Supporting_Document class

7.2.2.3.1 Direct superclass

Supporting_Document is a subclass of **Item** (see ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.3.2 Description of Supporting_Document

Supporting_Document is a class each instance of which models a supporting document, which might be a specification document, schema document or other reference document that provides specific information related to computable data.

7.2.2.3.3 Associations of Supporting_Document

As a subclass of **Item**, **Supporting_Document** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2). **Supporting_Document** has the following additional association:

— **computable_data_supporting_document** (see 7.2.3.2).

7.2.2.3.4 Attributes of Supporting_Document

The attributes of the **Supporting_Document** class are specified in Table 2.

Table 2 — Attributes of the Supporting_Document class

Attribute name	Multiplicity	Datatype	Description
document_role	0..1	**Text** (ISO/IEC 11179-3:2023, 6.2.12)	Definition: description of the role of the supporting document.
supporting_document	1..1	**Reference_Document** (ISO/IEC 11179-3:2023, 6.3.8)	Definition: a document that provides specific information related to computable data.
access_datetime	0..1	**Datetime** (ISO/IEC 11179-3:2023, 6.2.3)	Definition: date and time the document was accessed when it was used as a supporting document during the process the computable data are produced.

7.2.2.4 Computable_Data_Error class

7.2.2.4.1 Direct superclass

Computable_Data_Error is a subclass of **Item** (ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.4.2 Description of Computable_Data_Error

Computable_Data_Error is a class each instance of which models an error, which can be an empirical error, algorithmic error or some other type of error during the process by which computable data are produced by a pipeline.

The empirical error contains empirically determined values such as limits of detectability, false positives, false negatives, statistical confidence of outcomes, etc. This can be measured by running the algorithm on multiple data samples of the usability domain or through the use of carefully designed in-silico data.

The algorithmic error is descriptive of errors that originate by fuzziness of the algorithms, driven by stochastic processes, in dynamically parallelized multi-threaded executions, or in machine learning methodologies where the state of the machine can affect the outcome. This can be measured by taking a random subset of the data and re-running the analysis, or using some rigorous mathematical modeling of the accumulated errors and providing confidence values.

7.2.2.4.3 Associations of Computable_Data_Error

As a subclass of **Item**, **Computable_Data_Error** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2). **Computable_Data_Error** has the following additional association:

— **computable_data_error** (see 7.2.3.3).

7.2.2.4.4 Attributes of Computable_Data_Error

The attributes of the **Computable_Data_Error** class are specified in Table 3.

Table 3 — Attributes of the Computable_Data_Error class

Attribute name	Multiplicity	Datatype	Description
type	1..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: the type of the computable data error. Example: algorithmic error, empirical error, etc.
detail	1..1	**Text** (ISO/IEC 11179-3:2023, 6.2.12)	Definition: free text of detailed description of the computable data error.

7.2.2.5 Contributor class

7.2.2.5.1 Direct superclass

Contributor is a subclass of **Item** (ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.5.2 Description of Contributor

Contributor is a class each instance of which models a contributor, an entity that contributes to computable data.

A contributor can be a person, an organization, a software or a location, etc.

7.2.2.5.3 Associations of Contributor

As a subclass of **Item**, **Contributor** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2). **Contributor** has the following additional association:

— **computable_data_contributor** (see 7.2.3.4).

7.2.2.5.4 Attributes of Contributor

The attributes of the **Contributor** class are specified in Table 4.

Table 4 — Attributes of the Contributor class

Attribute name	Multiplicity	Datatype	Description
contributor_contribution	0..*	**Contribution** (7.2.4.1)	Definition: description of the type of contribution according to PAV ontology[3].

7.2.2.5.5 Constraint on Contributor

Each instance of the **Contributor** class shall only exist if at least one instance of the **item_designation** association of the parent **Item** class shall be instantiated.

7.2.2.6 Individual_Contributor class

7.2.2.6.1 Direct superclass

Individual_Contributor is a subclass of **Contributor** (see 7.2.2.5), which in turn is a subclass of **Item** (see ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.6.2 Description of Individual_Contributor

Individual_Contributor is a class each instance of which models an individual contributor, a person that contributes to computable data.

7.2.2.6.3 Associations of Individual_Contributor

As a subclass of **Item** and **Contributor**, **Individual_Contributor** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2) and **Contributor**'s associations (see 7.2.2.5.3).

7.2.2.6.4 Attributes of Individual_Contributor

The attributes of the **Individual_Contributor** class are specified in Table 5.

Table 5 — Attributes of the Individual_Contributor class

Attribute name	Multiplicity	Datatype	Description
contributor_affiliation	0..*	**Organization** (ISO/IEC 11179-3:2023, 6.3.3)	Definition: organization the individual contributor is affiliated with.
contributor_email	0..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: the email address of the individual contributor.
contributor_orcid	0..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: a persistent digital identifier for a person (the individual contributor) registered in ORCID.

7.2.2.7 Organization_Contributor class

7.2.2.7.1 Direct superclass

Organization_Contributor is a subclass of **Contributor** (see 7.2.2.5), which in turn is a subclass of **Item** (see ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.7.2 Description of Organization_Contributor

Organization_Contributor is a class each instance of which models an organization contributor, an organization that contributes to computable data.

7.2.2.7.3 Associations of Organization_Contributor

As a subclass of **Item** and **Contributor**, **Organization_Contributor** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2) and **Contributor**'s associations (see 7.2.2.5.3).

7.2.2.7.4 Attributes of Organization_Contributor

The attributes of the **Organization_Contributor** class are specified in Table 6.

Table 6 — Attributes of the Organization_Contributor class

Attribute name	Multiplicity	Datatype	Description
organization	0..1	**Organization** (ISO/IEC 11179-3:2023, 6.3.3)	Definition: organization that contributes to computable data.

7.2.2.8 Review class

7.2.2.8.1 Direct superclass

Review is a subclass of **Item** (see ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.8.2 Description of Review

Review is a class each instance of which models a review, which associates a reviewer and a description of the status of computable data in the review process.

7.2.2.8.3 Associations of Review

As a subclass of **Item**, **Review** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2). **Review** has the following additional association:

— **computable_data_review** (see 7.2.3.5).

7.2.2.8.4 Attributes of Review

The attributes of the **Review** class are specified in Table 7.

Table 7 — Attributes of the Review class

Attribute name	Multiplicity	Datatype	Description
review_date	0..1	**Date** (ISO/IEC 11179-3:2023, 6.2.4)	Definition: date of the review.
review_status	1..1	**Review_Status** (7.2.4.2)	Definition: current verification status in the review process of the computable data.
reviewer_name	1 :1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: the name of the reviewer.
reviewer_contribution	0..*	**Contribution**(7.2.4.1)	Definition: description of the type of contribution according to PAV ontology[3].
reviewer_affiliation	0.. *	**Organization** (ISO/IEC 11179-3:2023, 6.3.3)	Definition: organization the reviewer is affiliated with.
reviewer_email	0..*	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: the email address of the reviewer.
reviewer_orcid	0..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: a persistent digital identifier for a person (the reviewer) registered in ORCID.
reviewer_comment	0..1	**Text** (ISO/IEC 11179-3:2023, 6.2.12)	Definition: optional free text comment by a reviewer.

7.2.2.9 Computation_Step class

7.2.2.9.1 Direct superclass

Computation_Step is a subclass of **Item** (see ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.9.2 Description of Computation_Step

Computation_Step is a class each instance of which models a computation step, which is an individual tool (or a well defined and reusable script) used in a pipeline and can be executed sequentially or in parallel with other computation steps.

7.2.2.9.3 Associations of Computation_Step

As a subclass of **Item**, **Computation_Step** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2). **Computation_Step** has the following additional associations:

— **pipeline_compositon** (see 7.2.3.8);

— **computation_step_input** (see 7.2.3.9);

— **computation_step_output** (see 7.2.3.10);

— **computation_execution_environment** (see 7.2.3.11);

— **computation_step_prerequisite** (see 7.2.3.16);

— **computation_step_parameter** (see 7.2.3.17).

7.2.2.9.4 Attributes of Computation_Step

The attributes of the **Computation_Step** class are specified in Table 8.

Table 8 — Attributes of the Computation_Step class

Attribute name	Multiplicity	Datatype	Description
step_number	0..1	**Integer** (ISO/IEC 11179-3:2023, 6.2.5)	Definition: non-negative integer value representing the position of the computation step in a one-dimensional representation of the pipeline. While this document does not mandate any particular numbering schema, it is best practice to pick the most logically intuitive numbering system, and to keep numbering unique (i.e. do not repeat step numbers), such that each step has its own, unique integer. For example, a user may run one step at the same time as another step or analysis. In a case where each step pulls output from the same step, the steps may be step #3, and step #4, instead of number both as #3. The fact that they pull from the output of the same step (#2) can be detected from an association to the same instance of **Computation_Step_Prerequisite** class. Gaps are allowed (e.g. step 20 follows step 10).
version	0..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: version of the computation step used.
purpose	0..1	**Text** (ISO/IEC 11179-3:2023, 6.2.12)	Definition: description of the specific purpose of the computation step.

7.2.2.9.5 Constraint on Computation_Step

Each instance of the **Computation_Step** class shall only exist if at least one instance of the **item_designation** association of the parent **Item** class shall be instantiated.

7.2.2.10 Input_Output_Data class

7.2.2.10.1 Direct superclass

Input_Output_Data is a subclass of **Data_Set_Distribution** (see ISO/IEC 11179-33:2023, 7.2.2.5), which in turn is a subclass of **Item** (see ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.10.2 Description of Input_Output_Data

Input_Output_Data is a class each instance of which models input or output data of a computation step.

7.2.2.10.3 Associations of Input_Output_Data

As a subclass of **Item** and **Data_Set_Distribution**, **Input_Output_Data** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2) and **Data_Set_Distribution**'s associations (see ISO/IEC 11179-33:2023, 7.2.2.5.3). **Input_Output_Data** has the following additional associations:

— **computable_data_input** (see 7.2.3.6);

— **computable_data_output** (see 7.2.3.7);

— **computation_step_input** (see 7.2.3.9);

— **computation_step_output** (see 7.2.3.10).

7.2.2.10.4 Attributes of Input_Output_Data

The attributes of the **Input_Output_Data** class are specified in Table 9, The filename can be specified using **Designation.sign** of the parent **Item**.

Table 9 — Attributes of the Input_Output_Data class

Attribute name	Multiplicity	Datatype	Description
uri	1..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: URI of the data.
access_datetime	0..1	**Datetime** (ISO/IEC 11179-3:2023, 6.2.3)	Definition: date and time the data was accessed when it was used as input to the computation step.
creation_datetime	0..1	**Datetime** (ISO/IEC 11179-3:2023, 6.2.3)	Definition: date and time the data was created.
sha1_checksum	0:1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: SHA1 checksum of the data.

7.2.2.11 Computation_Execution_Environment class

7.2.2.11.1 Direct superclass

Computation_Execution_Environment is a subclass of **Item** (see ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.11.2 Description of Computation_Execution_Environment

Computation_Execution_Environment is a class each instance of which models a computation execution environment, and through its associations clearly describes the information about the environment needed to execute a computation step to the extent to which consistent results are obtained when produced repeatedly.

The information about the environment includes the platform, deployment, software configuration and running applications, etc.

7.2.2.11.3 Associations of Computation_Execution_Environment

As a subclass of **Item**, **Computation_Execution_Environment** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2). **Computation_Execution_Environment** has the following additional associations:

— **computation_execution_environment** (see 7.2.3.11);

— **computation_execution_script** (see 7.2.3.12);

— **computation_execution_software_prerequisite** (see 7.2.3.13);

— **computation_execution_environment_variable** (see 7.2.3.14);

— **computation_execution_external_data_endpoint** (see 7.2.3.15).

7.2.2.11.4 Attributes of Computation_Execution_Environment

The attributes of the **Computation_Execution_Environment class** are specified in Table 10.

Table 10 — Attributes of the Computation_Execution_Environment class

Attribute name	Multiplicity	Datatype	Description
platform	1..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: designation of an existing platform where this computation step was executed with a particular deployment of this platform. A platform can be a scientific platform such as Galaxy or HIVE or it can be a software package or apps that includes multiple algorithms and software.
script_driver	1..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: the mechanism to run an executable that can be launched to perform a sequence of commands described in the script. e.g. "manual", "shell".

7.2.2.12 Execution_Script class

7.2.2.12.1 Direct superclass

Execution_Script is a subclass of **Item** (see ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.12.2 Description of Execution_Script

Execution_Script is a class each instance of which models a script object. Script was used to perform computations. Script can be a script object, a computational service, or any other type of script.

7.2.2.12.3 Associations of Execution_Script

As a subclass of **Item**, **Execution_Script** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2). **Execution_Script** has the following additional associations:

— **computation_execution_script** (see 7.2.3.12).

7.2.2.12.4 Attributes of Execution_Script

The attributes of the **Execution_Script** class are specified in Table 11.

Table 11 — Attributes of the Execution_Script class

Attribute name	Multiplicity	Datatype	Description
filename	0..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: file name of the script.
uri	1..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: URI of the script.
access_datetime	0..1	**Datetime** (ISO/IEC 11179-3:2023, 6.2.3)	Definition: date and time the script was accessed when it was used for the computation execution.
sha1_checksum	0..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: SHA1 checksum of the script.

7.2.2.13 Software_Prerequisite class

7.2.2.13.1 Direct superclass

Software_Prerequisite is a subclass of **Item** (see ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.13.2 Description of Software_Prerequisite

Software_Prerequisite is a class each instance of which models a software prerequisite, which is the minimal necessary software or tool needed to successfully execute the computation.

7.2.2.13.3 Associations of Software_Prerequisite

As a subclass of **Item**, **Software_Prerequisite** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2). **Software_Prerequisite** has the following additional association:

— **computation_execution_software_prerequisite** (see 7.2.3.13).

7.2.2.13.4 Attributes of Software_Prerequisite

The attributes of the **Software_Prerequisite** class are specified in Table 12.

Table 12 — Attributes of the Software_Prerequisite class

Attribute name	Multiplicity	Datatype	Description
version	1..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: version of the software.
filename	0..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: file name of the software.
uri	1..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: URI of the software.
access_datetime	0..1	**Datetime** (ISO/IEC 11179-3:2023, 6.2.3)	Definition: date and time the software was accessed when it was used to execute the computation.
sha1_checksum	0..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: SHA1 checksum of the software.

7.2.2.13.5 Constraint on Software_Prerequisite

Each instance of the **Software_Prerequisite** class shall only exist if at least one instance of the **item_designation** association of the parent **Item** class shall be instantiated.

7.2.2.14 Environment_Variable class

7.2.2.14.1 Direct superclass

Environment_Variable is a subclass of **Item** (see ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.14.2 Description of Environment_Variable

Environment_Variable is a class each instance of which models a variable that is useful to configure the execution environment on the target platform.

7.2.2.14.3 Associations of Environment_Variable

As a subclass of **Item**, **Environment_Variable** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2). **Environment_Variable** has the following additional association:

— **computation_execution_environment_variable** (see 7.2.3.14).

7.2.2.14.4 Attributes of Environment_Variable

The attributes of the **Environment_Variable** class are specified in Table 13.

Table 13 — Attributes of the Environment_Variable class

Attribute name	Multiplicity	Datatype	Description
variable	1..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: specific variable for the computation execution environment.
value	1..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: variable value for the computation execution environment.

7.2.2.15 External_Data_Endpoint class

7.2.2.15.1 Direct superclass

External_Data_Endpoint is a subclass of **Item** (see ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.15.2 Description of External_Data_Endpoint

External_Data_Endpoint is a class each instance of which models an external data endpoint, which is minimal necessary domain specific external data sources accessed in order to successfully execute the computation.

7.2.2.15.3 Associations of External_Data_Endpoint

As a subclass of **Item**, **External_Data_Endpoint** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2). **External_Data_Endpoint** has the following additional association:

— **computation_execution_external_data_endpoint** (see 7.2.3.15).

7.2.2.15.4 Attributes of External_Data_Endpoint

The attributes of the **External_Data_Endpoint** class are specified in Table 14.

Table 14 — Attributes of the External_Data_Endpoint class

Attribute name	Multiplicity	Datatype	Description
url	1..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: The URL for the endpoint to be accessed.

7.2.2.15.5 Constraint on External_Data_Endpoint

Each instance of the **External_Data_Endpoint** class shall only exist if at least one instance of the **item_designation** association of the parent **Item** class shall be instantiated.

7.2.2.16 Computation_Step_Prerequisite class

7.2.2.16.1 Direct superclass

Computation_Step_Prerequisite is a subclass of **Item** (see ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.16.2 Description of Computation_Step_Prerequisite

Computation_Step_Prerequisite is a class each instance of which models a necessary package or resource needed for running a tool.

7.2.2.16.3 Associations of Computation_Step_Prerequisite

As a subclass of **Item**, **Computation_Step_Prerequisite** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2). **Computation_Step_Prerequisite** has the following additional association:

— **computation_step_prerequisite** (see 7.2.3.16).

7.2.2.16.4 Attributes of Computation_Step_Prerequisite

The attributes of the **Computation_Step_Prerequisite** class are specified in Table 15.

Table 15 — Attributes of the Computation_Step_Prerequisite class

Attribute name	Multiplicity	Datatype	Description
filename	0..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: file name of the package or resource.
uri	1..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: URI of package or resource.
access_datetime	0..1	**Datetime** (ISO/IEC 11179-3:2023, 6.2.3)	Definition: date and time the package or resource was accessed when it was used for running the tool.
sha1_checksum	0..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: SHA1 checksum of the package or resource.

7.2.2.17 Computation_Step_Parameter class

7.2.2.17.1 Direct superclass

Computation_Step_Parameter is a subclass of **Item** (see ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.17.2 Description of Computation_Step_Parameter

Computation_Step_Parameter is a class each instance of which models an optional parameter customizing the computational flow which can affect the output of the computation step.

NOTE Parameters can be custom to each kind of analysis and are tied to a particular pipeline implementation.

7.2.2.17.3 Associations of Computation_Step_Parameter

As a subclass of **Item**, **Computation_Step_Parameter** inherits **Item**'s associations (see ISO/IEC 11179-3:2023, 6.4.2.1.2). **Computation_Step_Parameter** has the following additional association:

— **computation_step_parameter** (see 7.2.3.17).

7.2.2.17.4 Attributes of Computation_Step_Parameter

The attributes of the **Computation_Step_Parameter** class are specified in Table 16.

Table 16 — Attributes of the Computation_Step_Parameter class

Attribute name	Multiplicity	Datatype	Description
parameter	1..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: specific variable for the computation step.
value	1..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: value (non-default)of the parameter.

7.2.3 Associations in the Computable_Data metamodel region

7.2.3.1 computable_data_pipeline association

The **computable_data_pipeline** association records the binding of one instance of the **Pipeline** class (see 7.2.2.2) to zero, one or more instances of the **Computable_Data** class (see 7.2.2.1).

This association registers the pipeline used for the generation of computable data.

7.2.3.2 computable_data_supporting_document association

The **computable_data_supporting_document** association records the binding of zero, one or more instances of the **Supporting_Document** class (see 7.2.2.3) to zero, one or more instances of the **Computable_Data** class (see 7.2.2.1).

This association registers the supporting documents of computable data, which are used to provide specification, validation schema, ontology, or other supporting information for the computable data.

7.2.3.3 computable_data_error association

The **computable_data_error** association records the binding of zero, one or more instances of the **Computable_Data_Error** class (see 7.2.2.4) to zero or one instance of the **Computable_Data** class (see 7.2.2.1).

This association registers an error with computable data during the process by which they are produced by the pipeline, which can be used to determine what range of input returns outputs that are within the tolerance level and therefore can be used to optimize algorithm.

7.2.3.4 computable_data_contributor association

The **computable_data_contributor** association records the binding of one or more instances of the **Contributor** class (see 7.2.2.5) to zero, one or more instances of the **Computable_Data** class (see 7.2.2.1).

This association registers the contributor(s) to the computable data.

7.2.3.5 computable_data_review association

The **computable_data_review** association records the binding of zero, one or more instances of the **Review** class (see 7.2.2.8) to zero or one instance of the **Computable_Data** class (see 7.2.2.1).

This association registers the review(s) of the computable data.

7.2.3.6 computable_data_input association

The **computable_data_input** association records the binding of one or more instances of the **Input_Output_Data** class (see 7.2.2.10) to zero, one or more instances of the **Computable_Data** class (see 7.2.2.1).

This association registers the input data to computable data.

7.2.3.7 computable_data_output association

The **computable_data_output** association records the binding of one or more instances of the **Input_Output_Data** class (see 7.2.2.10) to zero or one instance of the **Computable_Data** class (see 7.2.2.1).

This association registers the output data of computable data.

7.2.3.8 pipeline_composition association

The **pipeline_compositon** association records the binding of one or more instances of the **Computation_Step** class (see 7.2.2.9) to zero, one or more instances of the **Pipeline** class (see 7.2.2.2).

This association registers the inclusion of one or more computation steps in a pipeline.

7.2.3.9 computation_step_input association

The **computation_step_input** association records the binding of one or more instances of the **Input_Output_Data** class (see 7.2.2.10) to zero, one or more instances of the **Computation_Step** class (see 7.2.2.9).

This association registers the input data to a computation step.

7.2.3.10 computation_step_output association

The **computation_step_output** association records the binding of one or more instances of the **Input_Output_Data** class (see 7.2.2.10) to zero or one instance of the **Computation_Step** class (see 7.2.2.9).

This association registers the output data of a computation step.

7.2.3.11 computation_execution_environment association

The **computation_execution_environment** association records the binding of one instance of the **Computation_Execution_Environment** class (see 7.2.2.11) to zero, one or more instances of the **Computation_Step** class (see 7.2.2.9).

This association registers the execution environment of a computation step.

Accurate computation execution environment description will facilitate the reproducibility of computable data.

7.2.3.12 computation_execution_script association

The **computation_execution_script** association records the binding of one or more instances of the **Execution_Script** class (see 7.2.2.12) to zero, one or more instances of the **Computation_Execution_Environment class** (see 7.2.2.11).

This association registers the script of a computation execution.

7.2.3.13 computation_execution_software_prerequisite association

The **computation_execution_software_prerequisite** association records the binding of zero, one or more instances of the **Software_Prerequisite** class (see 7.2.2.13) to zero, one or more instances of the **Computation_Execution_Environment class** (see 7.2.2.11).

This association registers the necessary software or tool needed for a computation execution.

7.2.3.14 computation_execution_environment_variable association

The **computation_execution_environment_variable** association records the binding of zero, one or more instances of the **Environment_Variable** class (see 7.2.2.14) to zero, one or more instances of the **Computation_Execution_Environment class** (see 7.2.2.11).

This association registers the environment variable of a computation execution.

7.2.3.15 computation_execution_external_data_endpoint association

The **computation_execution_external_data_endpoint** association records the binding of zero, one or more instances of the **External_Data_Endpoint** class (see 7.2.2.15) to zero, one or more instances of the **Computation_Execution_Environment class** (see 7.2.2.11).

This association registers the external data endpoint of a computation execution.

7.2.3.16 computation_step_prerequisite association

The **computation_step_prerequisite** association records the binding of one or more instances of the **Computation_Step_Prerequisite** class (see 7.2.2.16) to zero, one or more instances of the **Computation_Step** class (see 7.2.2.9).

This association registers the necessary package or resource for running a computation step.

7.2.3.17 computation_step_parameter association

The **computation_step_parameter** association records the binding of zero, one or more instances of the **Computation_Step_Parameter** class (see 7.2.2.17) to zero, one or more instances of the **Computation_Step** class (see 7.2.2.9).

This association registers the parameter of a computation step.

7.2.4 Datatypes in the Computable_Data metamodel region

7.2.4.1 Contribution enumerated class

Contribution is an enumerated class. This enumerated class is used as the datatype for the **contribution** attribute of **Contributor** class (see 7.2.2.5) and **Review** class (see 7.2.2.8).

The values of the **Contribution** enumerated class are specified in Table 17.

Table 17 — Values in the Contribution enumerated class

Value	Description
“authoredBy”	Indicates an agent that originated or gave existence to the computable data.
“contributedBy”	Indicates an agent that provided any sort of help in conceiving the computable data.
“createdAt”	Indicates the geo-location of the agents when creating the computable data.
“createdBy”	Indicates an agent primary responsible for making the computable data representation.
“createdWith”	Indicates a software or tool used by the creator when making the computable data.
“curatedBy”	Indicates an agent responsible for shaping the expression in an appropriate format.
“derivedFrom”	Indicates a different resource that the computable data are derived from. If the computable data have the same content as the other resource, but have simply been transcribed to fit a different model (like XML -> RDF), use “importedFrom”. If the computable data were simply retrieved, use “retrievedFrom”. If the content of the computable data has however been further refined or modified, “derivedFrom” should be used.
“importedBy”	Indicates an entity responsible for importing the computable data.
“importedFrom”	Indicates an original source of imported computable data. NOTE Import means that the content has been preserved, but transcribed somehow, for instance to fit a different representation model by converting formats.
“providedBy”	Indicates an original provider of the resource the computable data was retrieved, imported or derived from.
“retrievedBy”	Indicates an entity responsible for retrieving the computable data from an external source.
“retrievedFrom”	Indicates a URI where the computable data have been retrieved from. NOTE Retrieval indicates that the computable data have the same representation as the original resource.
“sourceAccessedBy”	Indicates an entity that accessed or consulted (but did not retrieve, import or derive from) the computable data.
“sourceAccessedAt”	Indicates a URI where the source was accessed or consulted (but not retrieved, imported or derived from).

7.2.4.2 Review_Status enumerated class

Review_Status is an enumerated class. This enumerated class is used as the datatype for the **review_status** attribute of **Review** class (see 7.2.2.8).

The values of the **Review_Status** enumerated class are specified in Table 18.

Table 18 — Values in the Review_Status enumerated class

Value	Description
“proposed”	Indicates that the computable data has been submitted, and review is proposed.
“scheduled”	Indicates that the computable data has been submitted, and review is scheduled.
“in-review”	Indicates that the computable data has been submitted, and verification is under-way.
“approved”	Indicates that the computable data has been verified and reviewed.
“suspended”	Indicates that the computable data that was once valid is no longer considered valid.
“rejected”	Indicates that the computable data has been removed or rejected.

Annex A
(informative)

Mapping between IEEE 2791-2020 and this document

Table A.1 shows the mapping between IEEE 2791-2020 and this document.

Table A.1 — mapping between IEEE 2791-2020 and this document

No.	Domain and field in IEEE 2791-2020	Required	Class in this document	Attribute in this document	Note
1	object_id	Y			Mapped to “identifier” attribute of “Scoped_Identifier” class in 11179-3
2	spec_version	Y	*Supporting_Document* class (7.2.2.3)	document_role: Text [0..1] supporting_document: Reference_Document [1..1]	To ensure the generality of the metamodel, integrated *Cross_Reference* class, *Extension* class and *spec_version* attribute (which refers to the version of the BCO specification used to define the BCO) into one class: *Supporting_Document*
3	etag	Y	*Computable_Data* class (7.2.2.1)	etag: String [0..1]	
4	Provenance Domain	Y			Split into the following classes and attributes
4.1	name	Y			Mapped to “sign” attribute of “Designation” class in 11179-3
4.2	version	Y	*Computable_Data* class (7.2.2.1)	version: String [1..1]	
4.3	review	N	*Review* class (7.2.2.8)		
4.3.1	date	N	*Review class* (7.2.2.8)	review_date: Date [0..1]	
4.3.2	status	Y	*Review class* (7.2.2.8)	review_status: Review_Status [1..1]	
4.3.3	reviewer	Y			Split into the following attributes
4.3.3.1	name	Y	*Review class* (7.2.2.8)	reviewer_name: String [1..1]	
4.3.3.2	contribution	Y	*Review class* (7.2.2.8)	reviewer_contribution: Contribution [0..*]	
4.3.3.3	affiliation	N	*Review class* (7.2.2.8)	reviewer_affiliation: Organization [0..*]	
4.3.3.4	email	N	*Review class* (7.2.2.8)	reviewer_email: String [0..1]	
4.3.3.5	orcid	N	*Review class* (7.2.2.8)	reviewer_orcid: String [0..1]	
4.3.4	reviewer_comment	N	*Review class* (7.2.2.8)	reviewer_comment: Text [0..1]	
4.4	derived_from	N	*Computable_Data* class (7.2.2.1)	derived_from: String [0..1]	
4.5	obsolete_after	N	*Computable_Data* class (7.2.2.1)	obsolete_after: Datetime [0..1]	
4.6	embargo	N	*Computable_Data* class (7.2.2.1)	embargo_period: Datetime_Period [0..1]	
4.6.1	start_time	N	*Computable_Data* class (7.2.2.1)	embargo_period: Datetime_Period [0..1]	
4.6.2	end_time	N	*Computable_Data* class (7.2.2.1)	embargo_period: Datetime_Period [0..1]	
4.7	created	Y	*Computable_Data* class (7.2.2.1)	created_datetime: Datetime [0..1]	
4.8	modified	Y	*Computable_Data* class (7.2.2.1)	modified_datetime: Datetime [0..1]	
4.9	contributors	Y	*Contributor class* (7.2.2.5)		
4.9.1	name	Y			Mapped to “sign” attribute of “Designation” class in 11179-3
4.9.2	contribution	Y	*Contributor class* (7.2.2.5)	contributor_contribution: Contribution [0..*]	
4.9.3	affiliation	N	*Individual_Contributor class* (7.2.2.6)	contributor_affiliation: Organization [0..*]	
4.9.4	email	N	*Individual_Contributor class* (7.2.2.6)	contributor_email: String [0..1]	
4.9.5	orcid	N	*Individual_Contributor class* (7.2.2.6)	contributor_orcid: String [0..1]	

Table A.1 *(continued)*

No.	Domain and field in IEEE 2791-2020	Required	Class in this document	Attribute in this document	Note
4.10	license	Y	*Computable_Data* class (7.2.2.1)	licence: String [1..*]	
5	Usability Domain	Y	*Computable_Data* class (7.2.2.1)	usability: Text [0..*]	
6	Extension Domain	N	*Supporting_Document* class (7.2.2.3)	document_role: Text [0..1] supporting_document: Reference_Document [1..1]	To ensure the generality of the metamodel, integrated *Cross_Reference* class, *Extension* class and *spec_version* attribute (which refers to the version of the BCO specification used to define the BCO) into one class: *Supporting_Document*
7	Description Domain	Y			Split into the following classes and attributes
7.1	keywords	Y			Mapped to "sign" attribute of "Designation" class in 11179-3
7.2	xref	N	*Supporting_Document* class (7.2.2.3)		To ensure the generality of the metamodel, *Cross_Reference* class, *Extension* class and *spec_version* attribute (which refers to the version of the BCO specification used to define the BCO) have been integrated into one class: *Supporting_Document*
7.2.1	namespace	Y	*Supporting_Document* class (7.2.2.3)	supporting_document: Reference_Document [1..1]	
7.2.2	name	Y	*Supporting_Document* class (7.2.2.3)	supporting_document: Reference_Document [1..1]	
7.2.3	ids	Y	*Supporting_Document* class (7.2.2.3)	supporting_document: Reference_Document [1..1]	
7.2.4	access Time	Y	*Supporting_Document* class (7.2.2.3)	access_datetime: Datetime [0..1]	
7.3	platform	N	*Computation_Execution_Environment* class (7.2.2.11)	platform: String [1..1]	
7.4	pipeline_steps	Y	*Computation_Step* class (7.2.2.9)		
7.4.1	step_number	Y	*Computation_Step* class (7.2.2.9)	step_number: Integer [0..1]	
7.4.2	name	Y			Mapped to "sign" attribute of "Designation" class in 11179-3
7.4.3	description	Y	*Computation_Step* class (7.2.2.9)	purpose: Text [0..1]	
7.4.4	version	N	*Computation_Step* class (7.2.2.9)	version: String [0..1]	
7.4.5	prerequisite	N	*Computation_Step_Prerequisite* class (7.2.2.16)		
7.4.5.1	name	Y			Mapped to "sign" attribute of "Designation" class in 11179-3
7.4.5.2	uri	Y			Split into the following attributes
7.4.5.2.1	filename	N	*Computation_Step_Prerequisite* class (7.2.2.16)	filename: String [0..1]	
7.4.5.2.2	uri	Y	*Computation_Step_Prerequisite* class (7.2.2.16)	uri: String [1..1]	
7.4.5.2.3	access_time	N	*Computation_Step_Prerequisite* class (7.2.2.16)	access_datetime: Datetime [0..1]	
7.4.5.2.4	sha1_checksum	N	*Computation_Step_Prerequisite* class (7.2.2.16)	sha1_checksum: String [0..1]	
7.6	input_list	Y	*Input_Output_Data* class (7.2.2.10)		

Table A.1 *(continued)*

No.	Domain and field in IEEE 2791-2020	Required	Class in this document	Attribute in this document	Note
7.6.1	uri	Y			Split into the following attributes
7.6.1.1	filename	N			Mapped to “sign” attribute of “Designation” class in 11179-3
7.6.1.2	uri	Y	*Input_Output_Data* class (7.2.2.10)	uri: String [1..1]	
7.6.1.3	access_time	N	*Input_Output_Data* class (7.2.2.10)	access_datetime: Datetime [0..1]	
7.6.1.4	sha1_checksum	N	*Input_Output_Data* class (7.2.2.10)	sha1_checksum: String [0..1]	
7.7	output_list	Y	*Input_Output_Data* class (7.2.2.10)		
7.7.1	uri	Y			Split into the following attributes
7.7.1.1	filename	N			Mapped to “sign” attribute of “Designation” class in 11179-3
7.7.1.2	uri	Y	*Input_Output_Data* class (7.2.2.10)	uri: String [1..1]	
7.7.1.3	access_time	N	*Input_Output_Data* class (7.2.2.10)	access_datetime: Datetime [0..1]	
7.7.1.4	sha1_checksum	N	*Input_Output_Data* class (7.2.2.10)	sha1_checksum: String [0..1]	
8	Execution Domain	Y	*Computation_Execution_Environment* class (7.2.2.11)		
8.1	script	Y	*Execution_Script* class (7.2.2.12)		
8.1.1	uri	Y			Split into the following attributes
8.1.1.1	filename	N	*Execution_Script* class (7.2.2.12)	filename: String [0..1]	
8.1.1.2	uri	Y	*Execution_Script* class (7.2.2.12)	uri: String [1..1]	
8.1.1.3	access_time	N	*Execution_Script* class (7.2.2.12)	access_datetime: Datetime [0..1]	
8.1.1.4	sha1_checksum	N	*Execution_Script* class (7.2.2.12)	sha1_checksum: String [0..1]	
8.2	script_driver	Y	*Computation_Execution_Environment* class (7.2.2.11)	script_driver: String [1..1]	
8.3	software_prerequisites	Y	*Software_Prerequisite* class (7.2.2.13)		
8.3.1	name	Y			Mapped to “sign” attribute of “Designation” class in 11179-3
8.3.2	version	Y	*Software_Prerequisite* class (7.2.2.13)	version: String [1..1]	
8.3.3	uri	Y			Split into the following attributes
8.3.3.1	filename	N	*Software_Prerequisite* class (7.2.2.13)	filename: String [0..1]	
8.3.3.2	uri	Y	*Software_Prerequisite* class (7.2.2.13)	uri: String [1..1]	
8.3.3.3	access_time	N	*Software_Prerequisite* class (7.2.2.13)	access_datetime: Datetime [0..1]	
8.3.3.4	sha1_checksum	N	*Software_Prerequisite* class (7.2.2.13)	sha1_checksum: String [0..1]	
8.4	external_data_endpoints	Y	*External_Data_Endpoint* class (7.2.2.15)		

Table A.1 *(continued)*

No.	Domain and field in IEEE 2791-2020	Required	Class in this document	Attribute in this document	Note
8.4.1	name	Y			Mapped to “sign” attribute of “Designation” class in 11179-3
8.4.2	url	Y	*External_Data_Endpoint* class (7.2.2.15)	url: String [1..1]	
8.5	environment_variables	Y	*Environment_Variable* class (7.2.2.14)	variable: String [1..1] value: String [1..1]	
9	Parametric Domain	N	*Computation_Step_Parameter* class (7.2.2.17)		
9.1	step	Y			The computation_step_parameter association records the binding of zero, one or more instances of the Computation_Step_Parameter class (7.2.2.15) to zero, one or more instances of the Computation_Step class (7.2.2.7).
9.2	param	Y	*Computation_Step_Parameter* class (7.2.2.17)	parameter: String [1..1]	
9.3	value	Y	*Computation_Step_Parameter* class (7.2.2.17)	value: String [1..1]	
10	I/O domain	Y	*Input_Output_Data* class (7.2.2.10)		
10.1	Input subdomain	Y	*Input_Output_Data* class (7.2.2.10)		
10.1.1	uri	Y			Split into the following attributes
10.1.1.1	filename	N			Mapped to “sign” attribute of “Designation” class in 11179-3
10.1.1.2	uri	Y	*Input_Output_Data* class (7.2.2.10)	uri: String [1..1]	
10.1.1.3	access_time	N	*Input_Output_Data* class (7.2.2.10)	access_datetime: Datetime [0..1]	
10.1.1.4	sha1_checksum	N	*Input_Output_Data* class (7.2.2.10)	sha1_checksum: String [0..1]	
10.2	Output subdomain	Y	*Input_Output_Data* class (7.2.2.10)		
10.2.1	mediatype	Y			Mapped to “mediatype” attribute of “Data_Set_Distribution” class in 11179-33
10.2.2	uri	Y			Split into the following attributes
10.2.2.1	filename	N			Mapped to “sign” attribute of “Designation” class in 11179-3
10.2.2.2	uri	Y	*Input_Output_Data* class (7.2.2.10)	uri: String [1..1]	
10.2.2.3	access_time	N	*Input_Output_Data* class (7.2.2.10)	access_datetime: Datetime [0..1]	
10.2.2.4	sha1_checksum	N	*Input_Output_Data* class (7.2.2.10)	sha1_checksum: String [0..1]	
11	Error Domain	N	*Computable_Data_Error* class (7.2.2.4)		
11.1	Empirical error	N	*Computable_Data_Error* class (7.2.2.4)	type: String [1..1] detail: Text [1..1]	
11.2	Algorithmic error	N	*Computable_Data_Error* class (7.2.2.4)	type: String [1..1] detail: Text [1..1]	

Annex B
(normative)

Consolidated Class Hierarchy

Figure B.1 shows all classes specified in this document, or referenced from other parts of the ISO/IEC 11179 series, that participate in a class hierarchy. Classes that do not participate in a class hierarchy are not shown.

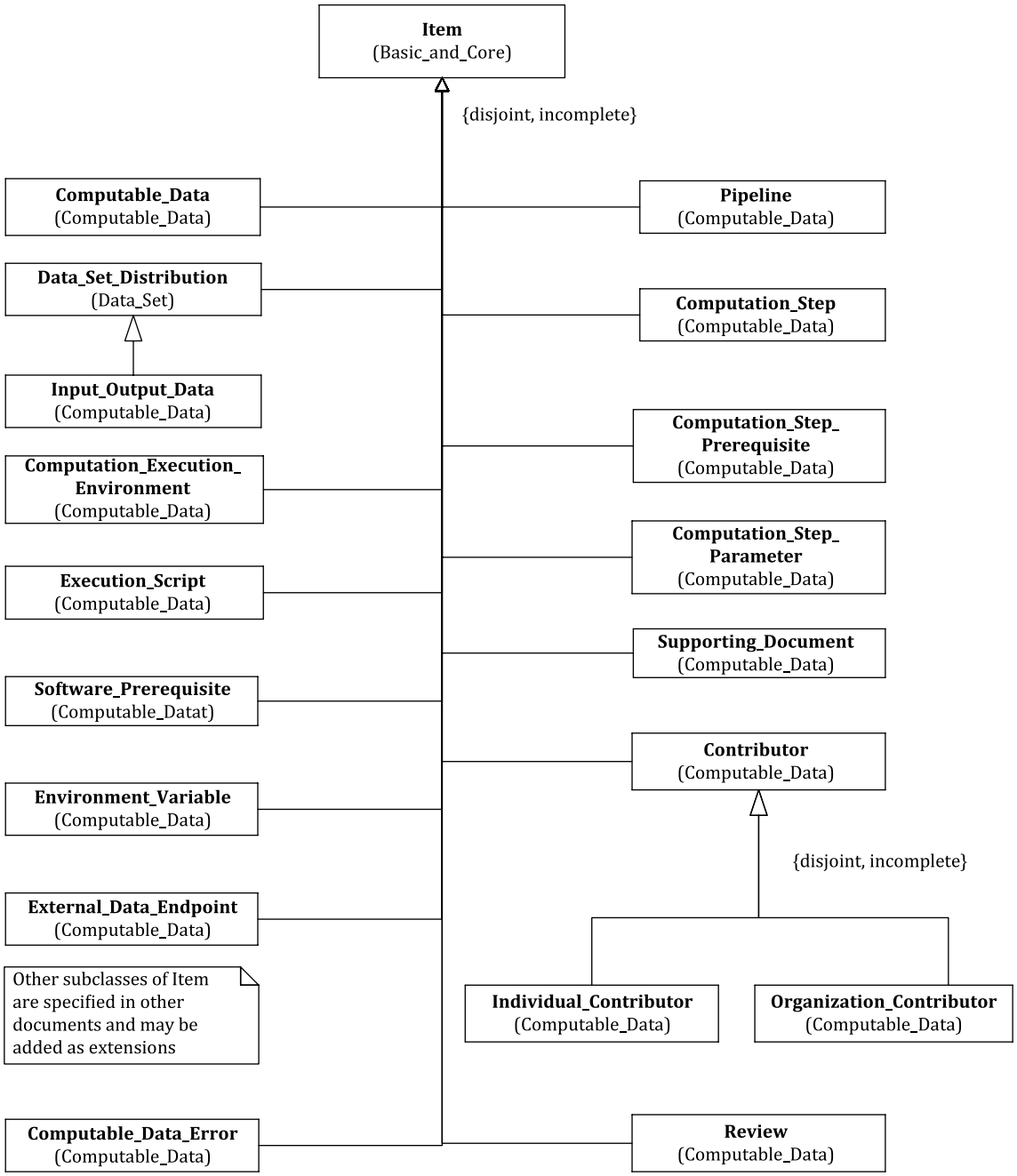

Figure B.1 — Consolidated Class Hierarchy

Standards or implementations which extend the metamodel in this document may extend this class hierarchy.

Annex C
(informative)

Examples of computable data registration

C.1 Overview

Two examples of the registration of computable data are provided in this Annex. The first example shows the registration of computable data in bioinformatics. The second example shows the registration of computable data in earth science.

C.2 Computable data example in bioinformatics

C.2.1 Overview of the example

In this example computable data are taken from the following website (accessed on 3 November 2021):

https://github.com/biocompute-objects/HIVE_metagenomics/blob/master/HIVE_metagenomics.json[8]

The computable data are represented in JSON format. The code is as follows:

```
{
    "bco_id": "https://github.com/biocompute-objects/HIVE_metagenomics/blob/master/HIVE_
metagenomics.json",
    "checksum": "ECD541AE0F61AAAAA1FAC14B2B08ABE18F610E1AA4677D54E89B292550F5058A",
    "bco_spec_version" : "https://w3id.org/biocompute/spec/v1.2",
    "provenance_domain": {
        "name": "Healthy human fecal metagenomic diversity",
        "version": "1.0.0",
        "review": [
            {
                "status": "approved",
                "reviewer_comment": "Approved by GW staff.",
                "reviewer": {
                    "name": "Charles Hadley King",
                    "affiliation": "George Washington University",
                    "email": "hadley_king@gwu.edu",
                    "contribution": ["curatedBy"],
                    "orcid": "https://orcid.org/0000-0003-1409-4549"
                }
            }
        ],
        "obsolete_after" : "2118-09-26T14:43:43-0400",
        "embargo" : {
            "start_time": "2000-09-26T14:43:43-0400",
            "end_time": "2000-09-26T14:43:45-0400"
        },
        "created": "2018-11-29T11:29:08-0500",
        "modified": "2018-11-30T11:29:08-0500",
        "contributors": [
            {
                "name": "Charles Hadley King",
                "affiliation": "George Washington University",
                "email": "hadley_king@gwu.edu",
                "contribution": ["createdBy", "curatedBy", "authoredBy"],
                "orcid": "https://orcid.org/0000-0003-1409-4549"
            },
            {
                "name": "Raja Mazumder",
                "affiliation": "George Washington University",
                "email": "mazumder@gwu.edu",
```

```
                "contribution": ["createdBy", "curatedBy", "authoredBy"],
                "orcid": "https://orcid.org/0000-0001-88238-9945"
            }
        ],
        "license": "https://spdx.org/licenses/CC-BY-4.0.html"
    },
    "usability_domain": [
        "Identify the most common organism present in a human [taxID:9606] fecal
[UBERON:0001988] sample, ",
        "Identify the general community composition of organisms in a human [taxID:9606] fecal
[UBERON:0001988] sample, ",
        "CensuScope is used to do a census of the composition of the read files. Based on a
user-defined threshold, organisms identified are used for alignment in the Hexagon alignment."
    ],
    "extension_domain":{
        "scm_extension": {
            "scm_repository": "https://github.com/biocompute-objects/HIVE_metagenomics",
            "scm_type": "git",
            "scm_commit": "e4620f642fb20557f6c679397696614305ed07b1",
            "scm_path": "biocompute-objects/HIVE_metagenomics",
            "scm_preview": "https://github.com/example/repo1/blob/
c9ffea0b60fa3bcf8e138af7c99ca141a6b8fb21/workflow/hive-viral-mutation-detection.cwl"
      }
    },
    "description_domain": {
        "keywords": [
            "metagenome",
            "metagenomic analysis",
            "fecal"
        ],
        "xref": [
            {
                "namespace": "uberon",
                "name": "Uber Anatomy Ontology",
                "ids": ["0001988"],
                "access_time": "2016-11-30T06:46-0500"
            },
            {
                "namespace": "taxonomy",
                "name": "Taxonomy",
                "ids": ["9606"],
                "access_time": "2016-11-30T06:46-0500"
            }
        ],
        "platform": ["hive"],
        " pipeline_steps": [
            {
                "step_number": 1,
                "name": "CensuScope",
                "description": "Detect taxonomic composition of a metagenomic data set.",
                "version": "1.3",
                "prerequisite": [
                    {
                        "name": "Filtered_NT_feb18_2016",
                        "uri": {
                            "uri": "https://hive.biochemistry.gwu.edu/genome/513957",
                            "access_time": "2016-11-30T06:46-0500"
                        }
                    }
                ],
                "input_list": [
                    {
                        "uri": "https://hive.biochemistry.gwu.edu/nuc-read/545722",
                        "access_time": "2016-11-30T06:46-0500"
                    },
                    {
                        "uri": "https://hive.biochemistry.gwu.edu/nuc-read/545721",
                        "access_time": "2016-11-30T06:46-0500"
                    }
                ],
                "output_list": [
                    {
```

```
                        "uri": "https://hive.biochemistry.gwu.edu/546223/
dnaAccessionBasedResult.csv",
                        "access_time": "2016-11-30T06:46-0500"
                    }
                ]
            },
            {
                "step_number": 2,
                "name": "HIVE-hexagon",
                "description": "Alignment of reads to a set of references",
                "version": "1.3",
                "input_list": [
                    {
                        "uri": "http://example.com/data/546223/dnaAccessionBased.csv",
                        "access_time": "2016-11-30T06:46-0500"
                    },
                    {
                        "uri": "https://hive.biochemistry.gwu.edu/nuc-read/545722",
                        "access_time": "2016-11-30T06:46-0500"
                    },
                    {
                        "uri": "https://hive.biochemistry.gwu.edu/nuc-read/545721",
                        "access_time": "2016-11-30T06:46-0500"
                    }
                ],
                "output_list": [
                    {
                        "uri": "https://hive.biochemistry.gwu.edu/546232/alCount-
Unalignedo524569-alCount--1.csv",
                        "access_time": "2016-11-30T06:46-0500"
                    }
                ]
            }
        ]
    },
    "execution_domain": {
        "script": [
            {
                "uri": {
                    "uri": "https://github.com/biocompute-objects/HIVE_metagenomics/blob/
master/driverHIVEmetagenomic.py"
                }
            }
        ],
        "script_driver": "shell",
        "software_prerequisites": [
            {
                "name": "CensuScope",
                "version": "albinoni.2",
                "uri": {
                    "uri": "http://example.com/dna.cgi?cmd=dna-screening&cmdMode=-",
                    "access_time": "2017-01-24T09:40:17-0500"
                }
            },
            {
                "name": "HIVE-hexagon",
                "version": "babajanian.1",
                "uri": {
                    "uri": "http://example.com/dna.cgi?cmd=dna-hexagon&cmdMode=-",
                    "access_time": "2017-01-24T09:40:17-0500"
                }
            }
        ],
        "external_data_endpoints": [
            {
                "name": "HIVE",
                "url": "https://hive.biochemistry.gwu.edu/dna.cgi?cmd=login"
            },
            {
                "name": "access to e-utils",
                "url": "https://eutils.ncbi.nlm.nih.gov/entrez/eutils/"
            }
```

```
    ],
    "environment_variables": {
        "key": "HOSTTYPE",
        "value" : "x86_64-linux"
    }
},
"parametric_domain": [
    {"param": "seed", "value": "14", "step": "2"},
    {"param":"minimum_match_len", "value": "66", "step": "2"},
    {"param": "divergence_threshold_percent", "value": "0.30", "step": "2"},
    {"param": "minimum_coverage", "value": "15", "step": "2"},
    {"param": "freq_cutoff", "value": "0.10", "step": "2"}
],
"io_domain": {
    "input_subdomain": [
        {
            "uri": {
                "filename": "Hepatitis C virus genotype 1",
                "uri": "https://www.ncbi.nlm.nih.gov/nuccore/22129792",
                "access_time": "2017-01-24T09:40:17-0500"
            }
        },
        {
            "uri": {
                "filename": "Hepatitis C virus type 1b complete genome",
                "uri": "https://www.ncbi.nlm.nih.gov/nuccore/5420376",
                "access_time": "2017-01-24T09:40:17-0500"
            }
        },
        {
            "uri": {
                "filename": "Hepatitis C virus (isolate JFH-1) genomic RNA",
                "uri": "https://www.ncbi.nlm.nih.gov/nuccore/13122261",
                "access_time": "2017-01-24T09:40:17-0500"
            }
        },
        {
            "uri": {
                "filename": "Hepatitis C virus clone J8CF, complete genome",
                "uri": "https://www.ncbi.nlm.nih.gov/nuccore/386646758",
                "access_time": "2017-01-24T09:40:17-0500"
            }
        },
        {
            "uri": {
                "filename": "Hepatitis C virus S52 polyprotein gene",
                "uri": "https://www.ncbi.nlm.nih.gov/nuccore/295311559",
                "access_time": "2017-01-24T09:40:17-0500"
            }
        },
        {
            "uri": {
                "filename": "HCV1a_drug_resistant_sample0001-01",
                "uri": "http://example.com/nuc-read/514682",
                "access_time": "2017-01-24T09:40:17-0500"
            }
        },
        {

            "uri": {
                "filename": "HCV1a_drug_resistant_sample0001-02",
                "uri": "http://example.com/nuc-read/514683",
                "access_time": "2017-01-24T09:40:17-0500"
            }
        }
    ],
    "output_subdomain": [
        {
            "mediatype": "text/csv",
            "uri": {
                "uri": "http://example.com/data/514769/dnaAccessionBased.csv",
                "access_time": "2017-01-24T09:40:17-0500"
```

```
                }
            },
            {
                "mediatype": "text/csv",
                "uri": {
                    "uri": "http://example.com/data/514801/SNPProfile*.csv",
                    "access_time": "2017-01-24T09:40:17-0500"
                }
            }
        ]
    },
    "error_domain": {
        "empirical_error": {
            "false_negative_alignment_hits": "<0.0010",
            "false_discovery": "<0.05"
        },
        "algorithmic_error": {
            "false_positive_mutation_calls_discovery": "<0.00005",
            "false_discovery": "0.005"
        }
    }
}
```


C.2.2 Registration of the example computable data

Figure C.1 and Figure C.2 show the registration of the example computable data using a UML object diagram.

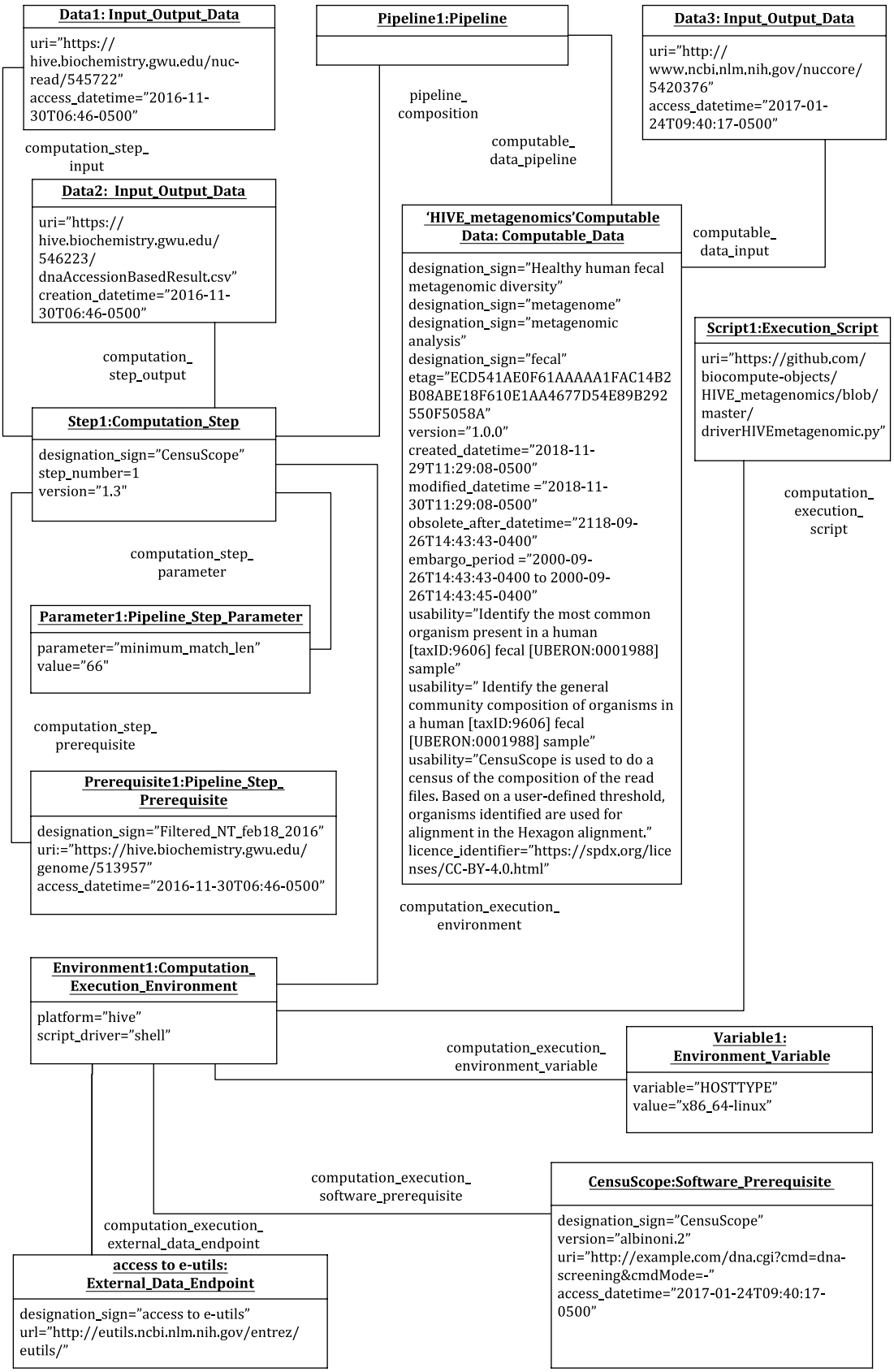

Figure C.1 — Registration of the ' HIVE_metagenomics ' computable data (1 of 2)

NOTE If there is more than one instance of the same class, not all instances are shown in the figure.

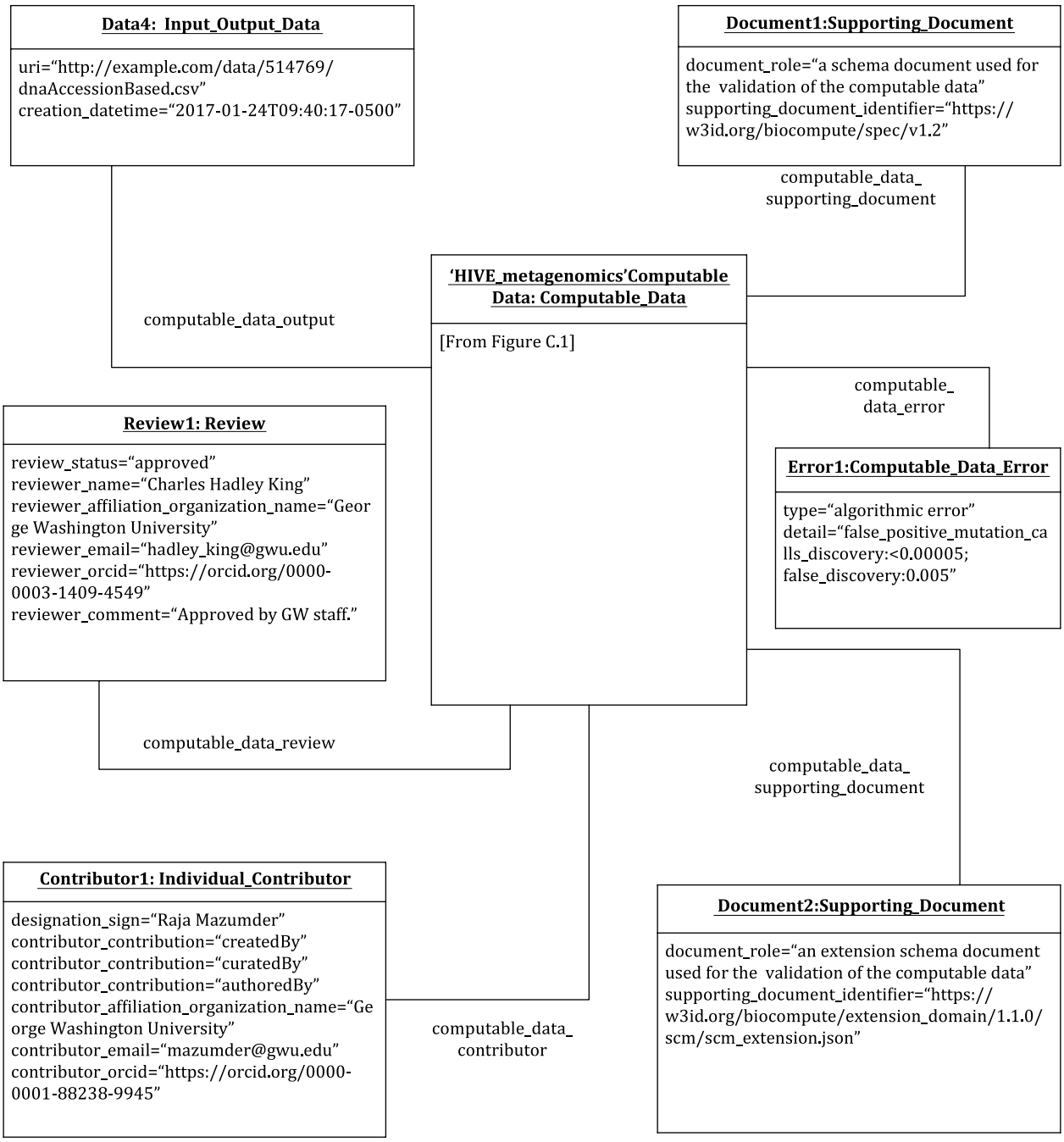

Figure C.2 — Registration of the ' HIVE_metagenomics ' computable data (2 of 2)

NOTE If there is more than one instance of the same class, not all instances are shown in the figure.

C.3 Computable data example in earth science

C.3.1 Overview of the example

In this example, computable data are taken from the following website (accessed on 3 May 2022):

https://reliance.rohub.org/overview?ac327c3a-5264-40a2-8c6e-1e8d7c4b37ef [9]

The computable data refers to the sea ice forecasting using IceNet notebook published in the Environmental Data Science book. It is encapsulated as a research object.

The input files, output files and the tools of this research object can be accessed and downloaded at the website above. The highlights include:

— Clone and access IceNet's codebase to produce seasonal Arctic sea ice forecasts using 3 out of 25 five pre-trained IceNet models downloaded from the Polar Data Centre;

— Forecast a single year, 2020, using IceNet's preprocessed environmental input data downloaded from a Zenodo repository;

— Visualise IceNet's seasonal ice edge predictions at 4- to 1-month lead times;

— Interactive plots comparing IceNet predictions against ECMWF SEAS5 physics-based sea ice concentration and a linear trend statistical benchmark.

C.3.2 Registration of the example computable data

Figure C.3 shows the registration of the example computable data using a UML object diagram.

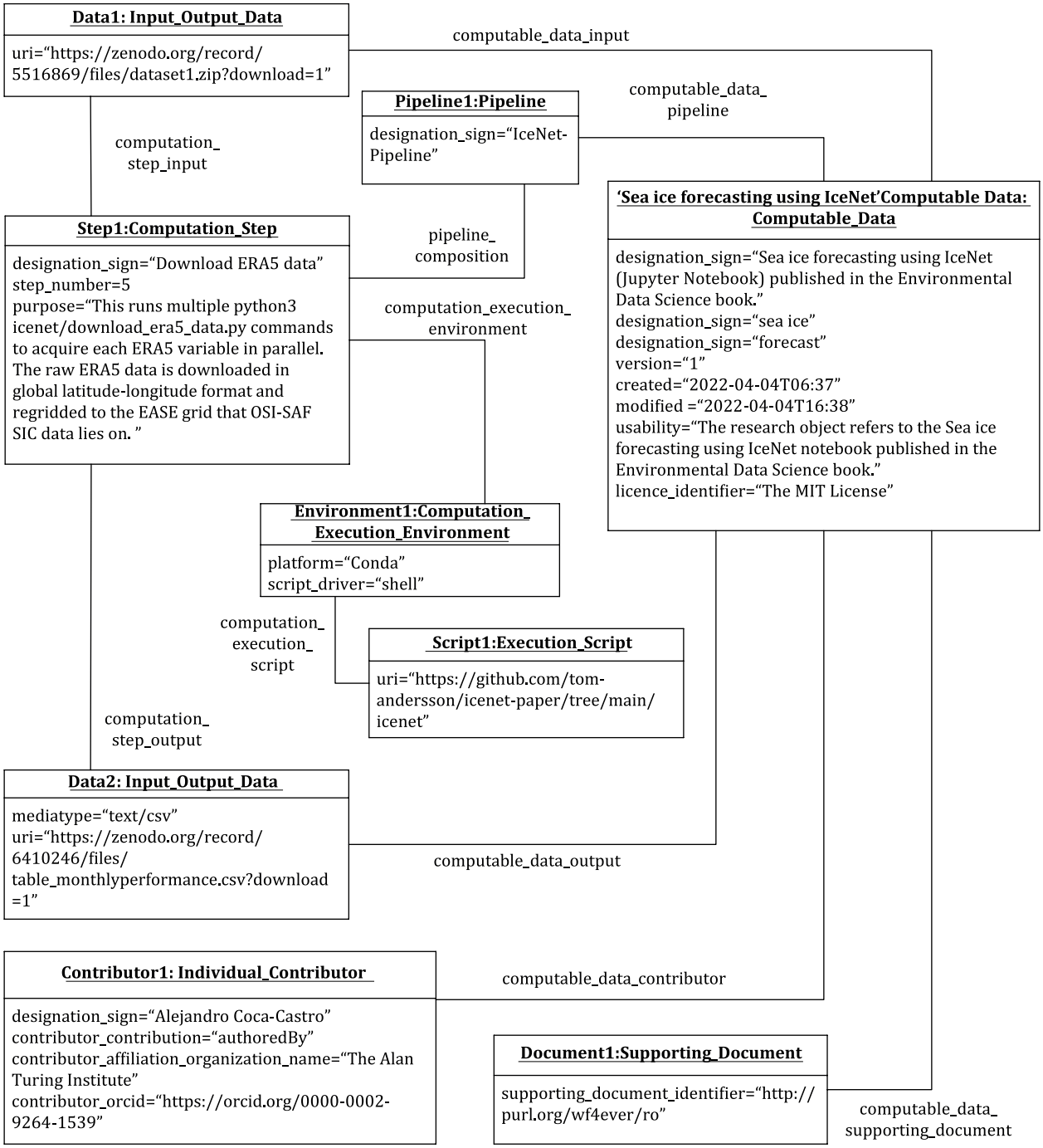

Figure C.3 — Registration of the ' Sea ice forecasting using IceNet' computable data

NOTE If there is more than one instance of the same class, not all instances are shown in the figure.

Bibliography

[1] IEEE 2791-2020, *IEEE Standard for Bioinformatics Analyses Generated by High-Throughput Sequencing (HTS) to Facilitate Communication*

[2] https://orcid.org/ (accessed 3 November 2021)

[3] https://pav-ontology.github.io/pav/ (accessed 3 November 2021)

[4] ISO/IEC 19505-1:2012, *Information technology — Object Management Group Unified Modeling Language (OMG UML) — Part 1: Infrastructure*

[5] ISO/IEC 19505-2:2012, *Information technology — Object Management Group Unified Modeling Language (OMG UML) — Part 2: Superstructure*

[6] https://tools.ietf.org/html/rfc7232 (accessed 3 November 2021)

[7] ATTRIBUTION C.C., 4.0 International https://spdx.org/licenses/CC-BY-4.0.html (accessed 3 November 2021)

[8] https://github.com/biocompute-objects/HIVE_metagenomics/blob/master/HIVE_metagenomics.json (accessed on 3 November 2021)

[9] COCA-CASTRO A., ANDERSSON T., BARLOW N., Sea ice forecasting using IceNet (Jupyter Notebook) published in the Environmental Data Science book." ROHub. Apr 03, 2022. https://w3id.org/ro-id/ac327c3a-5264-40a2-8c6e-1e8d7c4b37ef (accessed 3 November 2021)

[10] ISO/IEC 11179-6, *Information technology — Metadata registries (MDR) — Part 6: Registration*

[11] ISO/IEC 11179-33, *Information technology — Metadata registries (MDR) — Part 33: Metamodel for data set registration*

ICS 35.040.50
Price based on 41 pages

iso.org