INTERNATIONAL STANDARD

ISO 15926-10

First edition
2019-12

Industrial automation systems and integration — Integration of life cycle data for process plants including oil and gas production facilities —

Part 10:
Conformance testing

Reference number
ISO 15926-10:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

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 ISO documents 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).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

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.

This document was prepared by Technical Committee ISO/TC 184, *Automation systems and integration*, Subcommittee SC 4, *Industrial data*.

A list of all parts in the ISO 15926 series can be found on the ISO website.

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.

Introduction

This document demonstrates specified requirements of conformance to ISO 15926.

The target audiences for this document are as follows:

— users of a software product wishing to determine whether a software product deployed and/or a deliverable of the software product fulfils the specified requirements to conform to ISO 15926;

NOTE 1 Users typically include, but are not limited to, plant owners, project management contractors, front end engineering design contractors, engineering procurement construction contractors, original equipment manufacturing suppliers, catalogue providers, commissioning engineers, information technology engineers, and information management engineers.

— implementers wishing to determine whether a software product developed and/or a deliverable of the software product fulfils the specified requirements to conform to ISO 15926.

NOTE 2 Implementers include, but are not limited to, software engineers working for commercial software product development companies, and software engineers working on industrial businesses using their developed software product.

Users of this document are expected to have an understanding of conceptual data models, of the ISO 15926 series, of the ISO/TS 18876 series and of ISO/IEC/IEEE 15288.

Industrial automation systems and integration — Integration of life cycle data for process plants including oil and gas production facilities —

Part 10: Conformance testing

1 Scope

This document defines the principles and methods for conformance testing of software implementations of ISO 15926.

It provides guidance for developing test cases and testing procedures that cover the requirements specified in the ISO 15926 series and in different industry usage contexts, e.g. data exchange, use of reference data libraries and interface services.

This document provides guidance in addition to the conformance in the parts.

NOTE 1 Guidance on conformance ISO 15926 testing of complex scenarios which represent integrated interoperability is outside the scope of this document.

NOTE 2 Guidance on the development of software that supports the way of file exchange in the simple scenario which represents unified interoperability is outside the scope of this document.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

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

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

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

3.1
conformance testing
process to determine whether an implementation meets the requirements of a standard

3.2
specified requirement
need or expectation that is stated

3.3
functional approach
series of three functions that satisfy a need or demand for a demonstration that *specified requirements* (3.2) are fulfilled

Note 1 to entry: The three functions are selection, determination, and review and attestation.

3.4
software implementation layer
conceptual model that characterizes the software implementation

3.5
information exchange layer
particular conceptual model based on the *software implementation layers* (3.4) that characterizes the information exchange between computing system

3.6
ontology
formal statement of an understanding of the world

Note 1 to entry: An ontology can be represented in any language. It need not be represented in a language specifically designed for ontologies, such as OWL. An ontology can have different representations.

Note 2 to entry: An ontology does not specify what data shall be recorded about the world.

Note 3 to entry: The ontology defined by this part of ISO 15926 is principally concerned with the world outside a computer system.

[SOURCE: ISO/TS 15926-12:2018, 3.1.3]

4 Principles and methods

4.1 Conformance testing

Conformance testing of software implementations of ISO 15926 (conformance to the ISO 15926 series) shall be performed according to the functional approach, which is described in ISO/IEC 17000.

The testing can be done by first, second and/or third party.

NOTE The adoption of ISO/IEC Guide 60 can build the credibility of conformance testing.

4.2 Conformance to the ISO 15926 series

4.2.1 Conformance to ISO 15926-1

Conformance testing of software implementations of ISO 15926 means conformance to ISO 15926-1, if and only if all the requirements stated by the users of a software product and/or the implementer are fulfilled. A statement of conformance shall be prepared to state all means of communicating that the fulfilment of specified requirements has been successfully demonstrated.

ISO 15926-1 describes the activity, activity analysis, and modelling methods. The ISO 15926-1 activity model is shown in Figure 1.

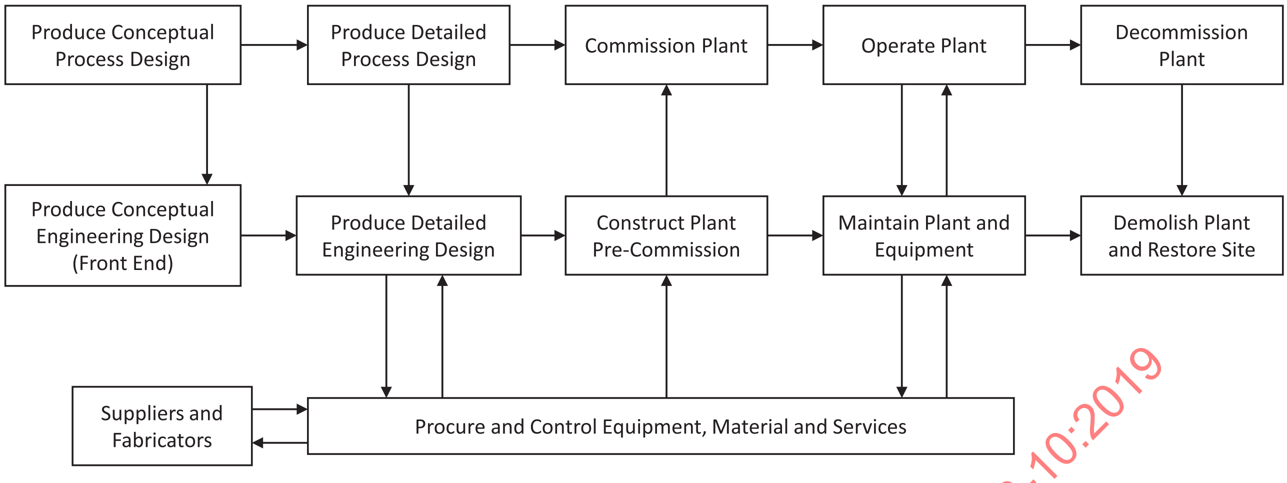

Figure 1 — ISO 15926-1 activity model

4.2.2 ISO 15926-1 conformance scenarios

Within the ISO 15926-1 activity model, there is a domain and range of scenarios, each of which results in various complexity of the software implementations of ISO 15926:

— scenario of handover, including sending and receiving files conforming to parts of ISO 15926;

— scenario of integration including using software programs that support the full span of design, engineering, construction, operation, and maintenance.

For each scenario, the following information shall be collected and recorded, but not limited to:

— overview (of scenario conformance);

— general (of the scenario of the annex and systems involved);

— procedure (steps of the workflow of handover or integration);

— aspects to be validated (verification);

— detailed validation (verification);

— not in scope (define what is not included in the workflow steps but useful to know).

Conformance testing of software implementations of ISO 15926 shall be executed in accordance with the scenario. Moreover, by applying the functional approach, the scenario shall demonstrate that the commissioned system fulfils specified requirements.

To perform the functional approach, the following activities, but not limited to, shall be conducted:

— selection (of information): planning and preparation activities to collect and produce all information and input needed for the subsequent determination functions;

— determination (gather evidence): developing complete information regarding fulfilment of the specified requirements by the object of conformity assessment or its sample;

— review (of evidence) and attestation (to conformity): constituting the final stage of checking before taking the important decision as to whether the purpose of conformity assessment has been reliably demonstrated to fulfil the specified requirements.

The specific application of the functional approach shall be documented by the conformance testing plan.

An appropriate set of validation and/or verification rules shall be selected and applied by referring to Annexes A, B, C, D, E and F. The candidate set of verification rules shall include, but not be limited to, the following (see Figure 3):

— all applications at layer 1 shall adopt Annex A;

— all applications at layer 2 shall adopt Annex B;

— an applications at layer 3 that claims conformance to ISO/TS 15926-7 and ISO/TS 15926-8 shall refer to Clause 5 for the description of the methodology and adopt the conformance rules given in Annex C;

— an application at layer 3 that claims conformance to ISO/TS 15926-11 shall refer to Clause 6 for the description of the methodology and adopt the conformance rules given in Annex D;

— an application at layer 3 that claims conformance to ISO/TS 15926-12 shall adopt Annex E;

— an application at layer 3 that claims conformance to ISO 15926-13 shall adopt Annex F.

NOTE This selection of annexes can be interpreted as conformance clauses used in the ISO 10303 series. Moreover, the particular selection of annex to test the conformance can be described as the conformance class of the particular part of the ISO 15926 series.

4.2.3 Conformance testing in accordance with ISO 15926-1

The software implementation of ISO 15926 consists of four layers. For each selected layer, the specific functional approach in accordance with the following shall be applied.

— Role and scope of the software implementation (layer 1): This is to define the “what” portion of the software implementation of ISO 15926 in terms of the role and scope. Based on the activity model in ISO 15926-1, actual placement of the role and scope will be identified. The conformance testing plan or its equivalent shall be documented. Role and scope shall be examined by referring to Annexes A and B.

— Content of the software implementation (layer 2): The particular content of the software implementation of ISO 15926 shall be defined based on information defined in above layer 1. It is the layer where the particular information and/or product model(s) referring the entities and relationships of ISO 15926-2 and accompanying domain and range of ISO/TS 15926-4 or relent reference data library (RDL) based on ISO/TS 15926-6 (the whole integration model) shall be defined and assessed. Use of context shall be examined by referring Annexes A and B.

— Semantics of the software implementation (layer 3): The particular semantics of the software implementation shall be designed by selecting the preferred combination of the part(s) of the ISO 15926 series. Both the source and target application shall be mapped to the particular information and/or product model(s) defined in the above layer 2. Use of semantic shall be examined by referring to Annexes C to F.

— Syntax and storage of the software implementation (layer 4): The syntax and storage shall be defined by the actual implementation of language and storage of the preferred part(s) of the ISO 15926 series. Conformance testing can test, but not limited to, correct use of syntax, URI, and model which are defined in selected implementation languages.

At each layer, conformance testing of a software implementation of ISO 15926 shall be conducted to see if a particular commissioned system meets the user requirements. Furthermore, depending on the software implementation layer, conformance rules defined in accompanying annexes shall be selected and used to determine the conformance, and reviewed and attested the results as a part of the conformance. See Table 1 for an overall summary.

Table 1 — Software implementation layer of ISO 15926 and conformance testing

Software implementation layer of ISO 15926	What required and what rules to be selected	What to be validated and/or verified
Role and scope (Layer 1)	Business Requirements and selecting handover or integration schema.	Validate quality (usability, completeness, and consistency) of a handover or integration schema.
Content (Layer 2)	RDL content management requirements and selecting relevant Reference Data Items for handover or for an integration scheme.	Verify whether the Reference Data Items are indeed available for the selected handover or integration scheme.
Semantics (Layer 3)	Semantic requirements and selecting appropriate ISO 15926 part(s) for handover or integration.	Verify correctness of the semantic interpretation of data for handover or integration.
Syntax and storage (Layer 4)	Technological requirements and selecting appropriate serialization and data for handover or integration technology.	Verify correctness serialization and data for handover or integration technology

The layer model is valid both for integrated interoperability and for unified interoperability, as described in ISO 11354-1. The integrated approach assures consistency and coherence of the interoperating subsystems by focusing on the (software) components that need to interact. These components are then designed and implemented using a common standard so that interoperability is seen as a designed-in quality. Interoperation between these various components is therefore obtained a priori without any interfacing effort. The choice of interoperability has a significant impact on the practical interpretation of Table 1.

This document deals with the so-called “simple scenario” of exchange of a file which follows the unified interoperability approach, as described in ISO 11354-1. The unified interoperability approach uses a common meta-level structure that shall be identified and detailed applicable for the participating parties. This structure shall provide a means for semantic equivalence to allow mapping between entities. Using this meta-level structure, a translation between the constituent entities is then possible.

More complex scenarios aiming at integrated interoperability, are not covered in this document.

4.3 Simple scenario

4.3.1 Overview

The general process of data exchange in the context of ISO 15926 is explained from a practical point of view of conformance testing of the software implementation to ISO 15926, taking in count the technology available today. This subclause is about data exchange in terms of physically exchanging a file between two parties to exchange explicit, unambiguous asset management information. Development of software that supports this way of file exchange is outside the scope of this document.

4.3.2 Introduction

As an example, the exchange of data in the creation process of an asset owner with a contractor is used in the context of a unified interoperability approach. The principle used in this example is shown in Figure 2. The handover is represented by an envelope containing the payload file.

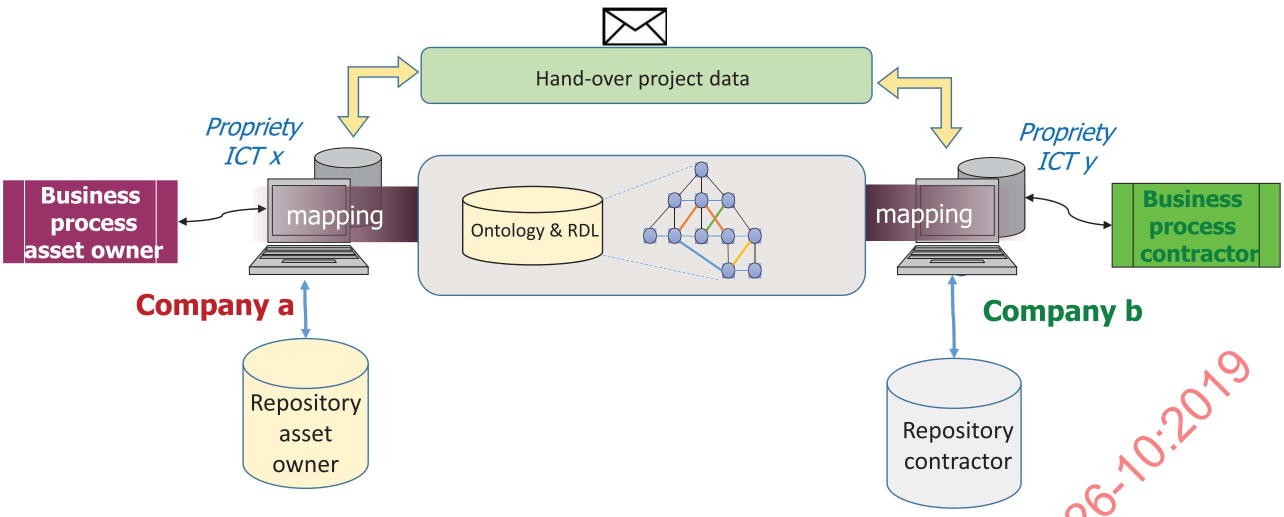

Figure 2 — Principle of data exchange on a commonly shared ontology and RDL

4.3.3 Process steps of the workflow of exchange data

Within this document, a systematic approach using four software implementation layers is presented in Figure 3, in order to organize the exchange process.

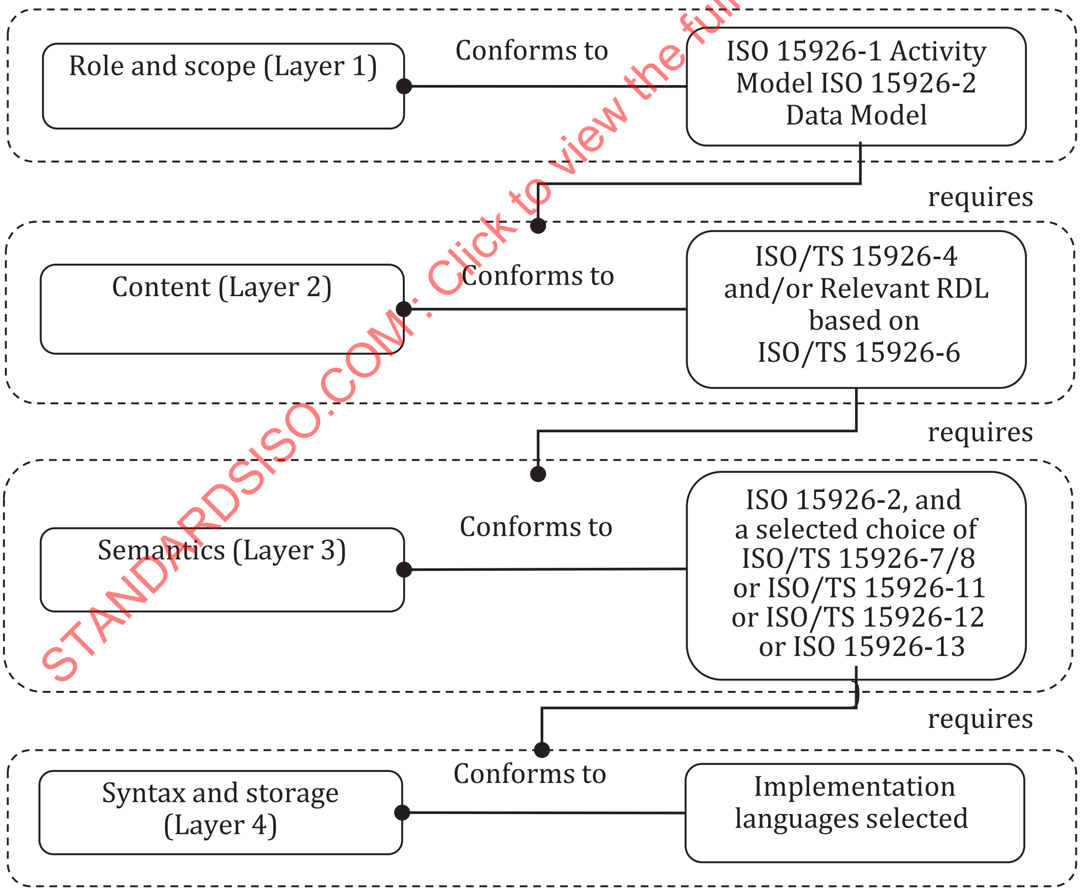

Figure 3 — Layers within the process of data exchange

The four layers are shown in Figure 3 are as follows:

— role and scope of the data exchange (the red arrows in Figure 4);

— content: definitions of the objects that can be found in the exchange file, classified according to the shared RDL;

— semantics (meaning) of the data exchanged defined by the part or parts of ISO 15926 agreed on;

— syntax and storage: the method of serialization and syntax used on the data level.

A step-by-step enumeration of the tasks carried out during a process has been extracted for a chosen scenario. Moreover, it shall fall into some of the following activities executed by the asset owner either the contractor where specific entities, e.g. objects in the context of a specific plant or project shall be determined. This process can be referred to as “role and scope.”

Furthermore, once a relevant entity is identified, the relevant RDL items shall be selected.

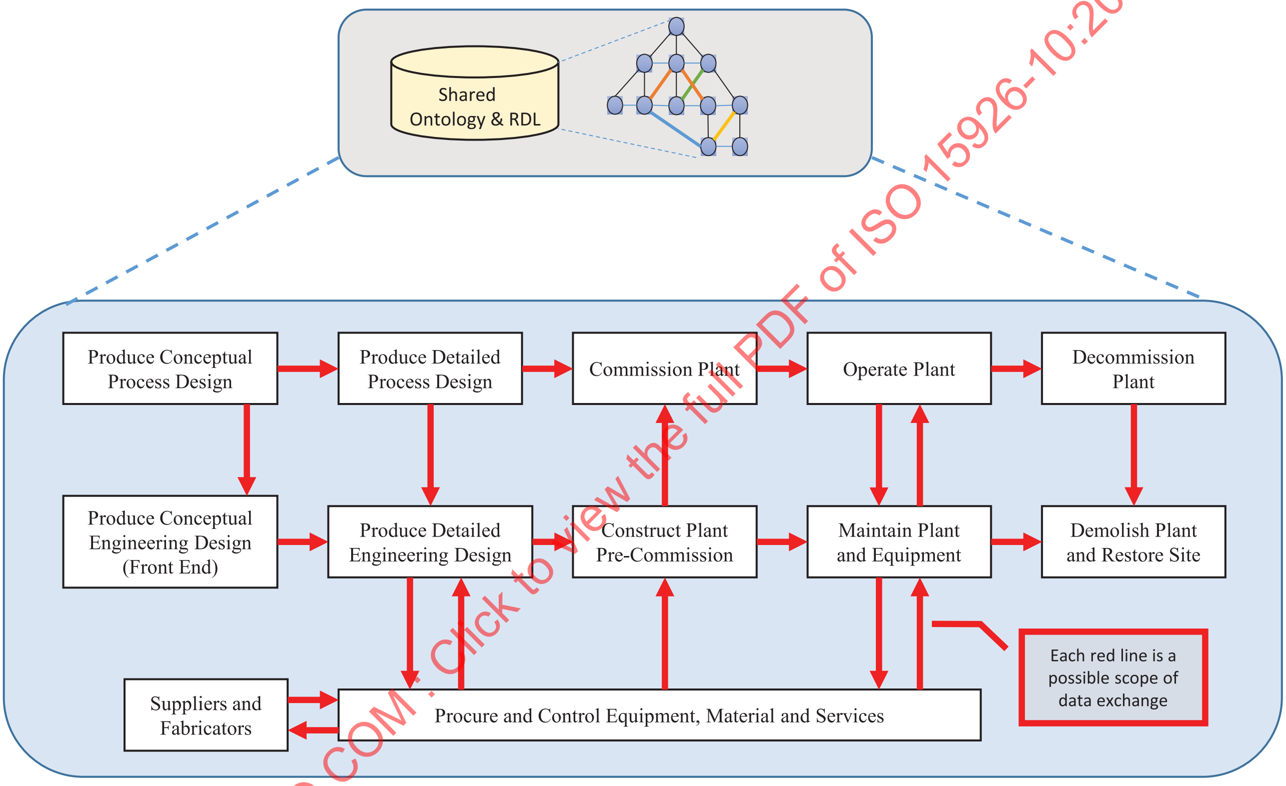

Figure 4 — Activity model as a basis for the selection of role and scope of data exchange

4.3.4 Aspects to be validated

In Figure 5, a scenario is presented where the asset owner, designing a process plant, exchange data with the contractor about this plant in the context of a contract between the asset owner and the contractor about delivering the design made by the contractor to the asset owner.

The total design of the process plant is stored in a repository owned by the asset owner. The contractor does have its repository to store and maintain its design, including all information, not directly relevant for the asset owner.

Within the project, a shared ontology and RDL (both on class level) is agreed and represents the semantics of the design to be stored and exchanged on the data level (content layer in Figure 3).

All objects within both the repository of the asset owner and the contractor are classified as a class in the shared RDL and are identified by commonly used identifiers (UID’s, recognized by both plant owner and contractor).

The moment the asset owner has the essential information in its repository to deliver the contractor the context for the design scope, the asset owner extracts this information as a data set from its repository.

The asset owner transforms the dataset according to the semantic representation method (a specific part or coherent set of ISO 15926 parts) and verifies the data set against the classes, rules, and constraints as captured in the ontology and RDL (semantics layer in Figure 3).

The asset owner packs this dataset in a digital envelope according to the serialization method as specified in the handover specification and sends this envelope to the contractor.

The contractor receives the envelope and checks whether it can unpack the data set, according to the specified methods in the handover specification. The contractor verifies the data set against the classes, allowed relations, rules, and constraints as captured in the ontology and RDL.

The data set is imported in its repository, so contractor now knows the context of its design scope in terms of the class of systems or equipment that needs to be designed and corresponding identifiers of already defined systems and equipment with their classes, within the repository of the asset owner.

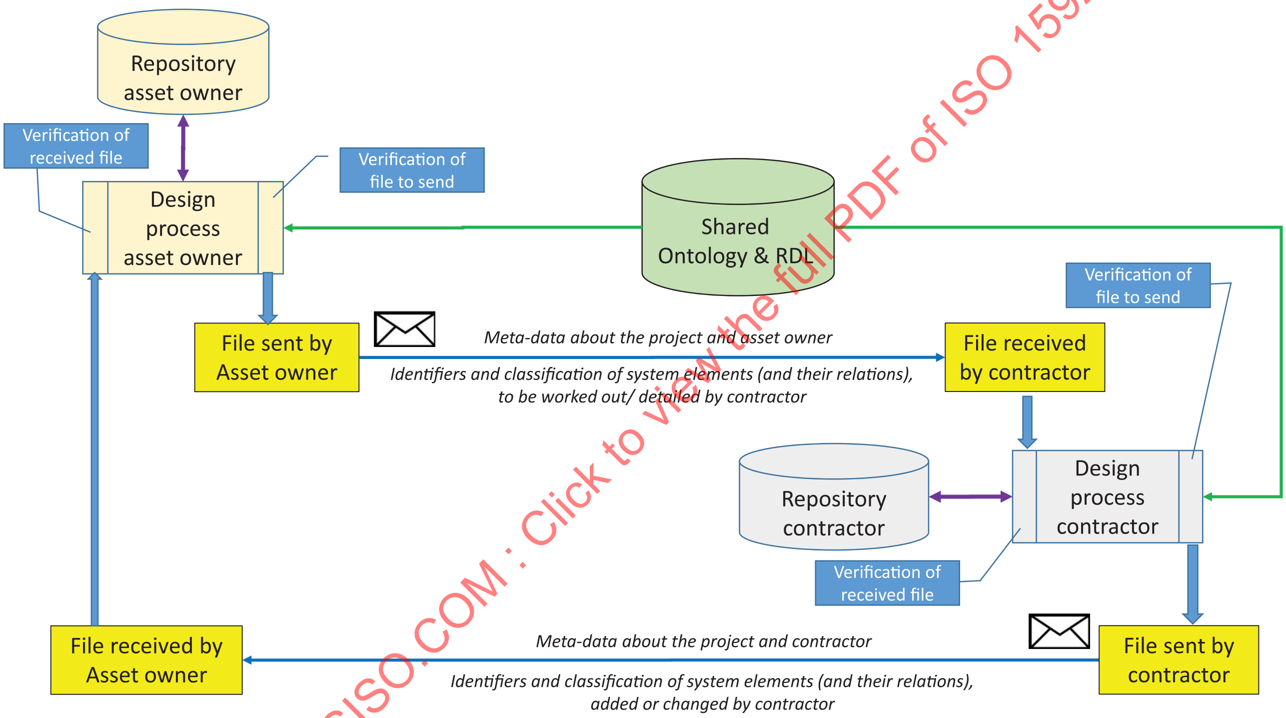

Figure 5 — Bi-directional data exchange scenario between asset owner and contractor

Furthermore, two additional scenarios for the metadata of the project (name and identifier of the asset owner, name, identifier of the plant, and date sent) can be realized using:

— separate header file in the envelope;

— integration of this metadata in the payload file (metadata model according to the ontology).

The above statement leads to the following two options:

— scenario needs to be defined in the handover specification;

— asset owner can decide to include the ontology and RDL files within the envelope.

The contractor adds equipment with identifiers and specifications, classified according to the ontology and RDL classes and herein defined relations to the baseline as delivered by the asset owner.

The moment the contractor wants to exchange a baseline of the contractor's design with the asset owner, the contractor extracts the relevant data set (the set of objects the contractor wants to deliver to the plant owner) from the contractor's repository

The contractor transforms the dataset according to the semantic representation method (a specific part or coherent set of ISO 15926 parts) and verifies the data set against the classes, rules and constraints as captured in the ontology and RDL and verifies the data set against the classes, allowed relations, rules and constraints as captured in the ontology and RDL.

Then the contractor packs the dataset into the envelope, including metadata about the delivery according to the agreed scenario in the handover specification.

4.3.5 Detailed validation

The process steps follow the layers as defined in this document, shown in Figure 6.

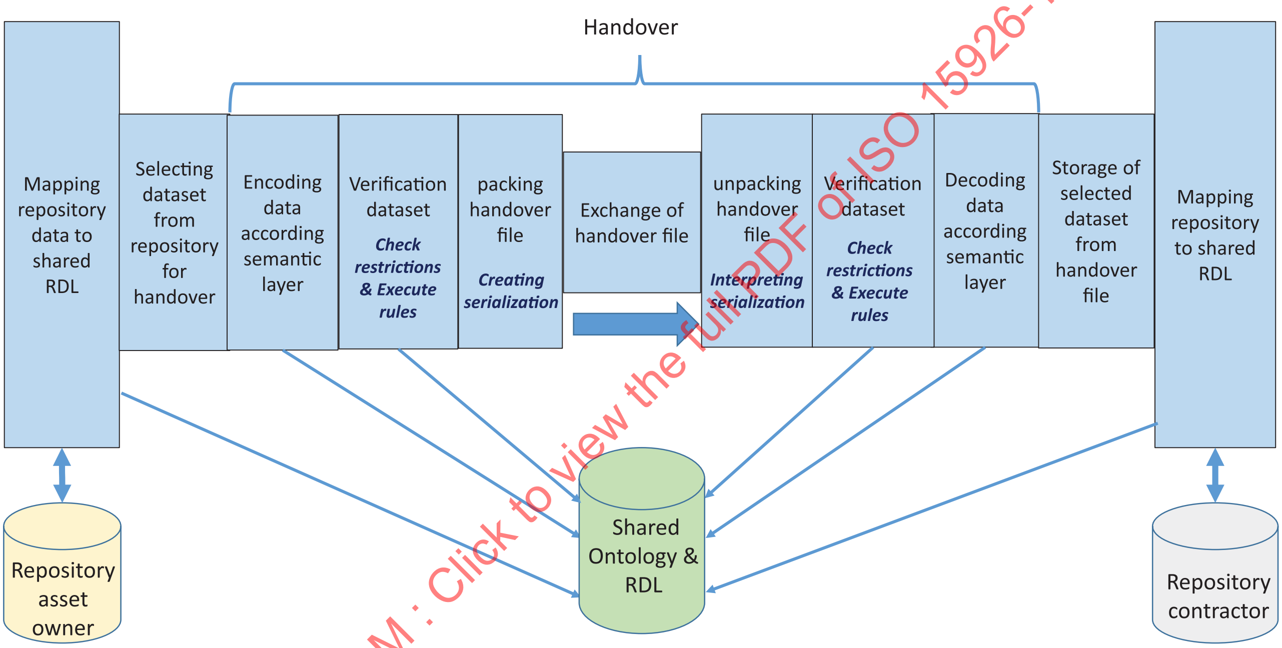

Figure 6 — Process of exchange seen from the asset owner to the contractor

Specific rules and constraints may be specified in the handover specific to the action when not applied in the ontology. The chosen serialization method should also be specified in the handover specification.

NOTE Alternatively, comparison between the scenario and the scenarios in annexes reveals what rules to be referred to and added or deleted from the rule set defined in the chosen annex.

In Figure 7, the layers as shown in Figure 6 are mapped to the corresponding process steps of Figure 5. For each step, there should be a dedicated clause in information handover specification to specify in detail at the layer level. The clause includes, but is not necessarily limited to:

— ontology and RDL to use, including its version;

— which semantic principle needs to be used for encoding the information (e.g. the selected choice of ISO/TS 15926-7 or ISO/TS 15926-8, ISO/TS 15926-11, ISO/TS 15926-12, ISO 15926-13);

— which serialization needs to be used (XML, Turtle, etc.).

This is respectively presented as syntax and storage layer in Figure 3.

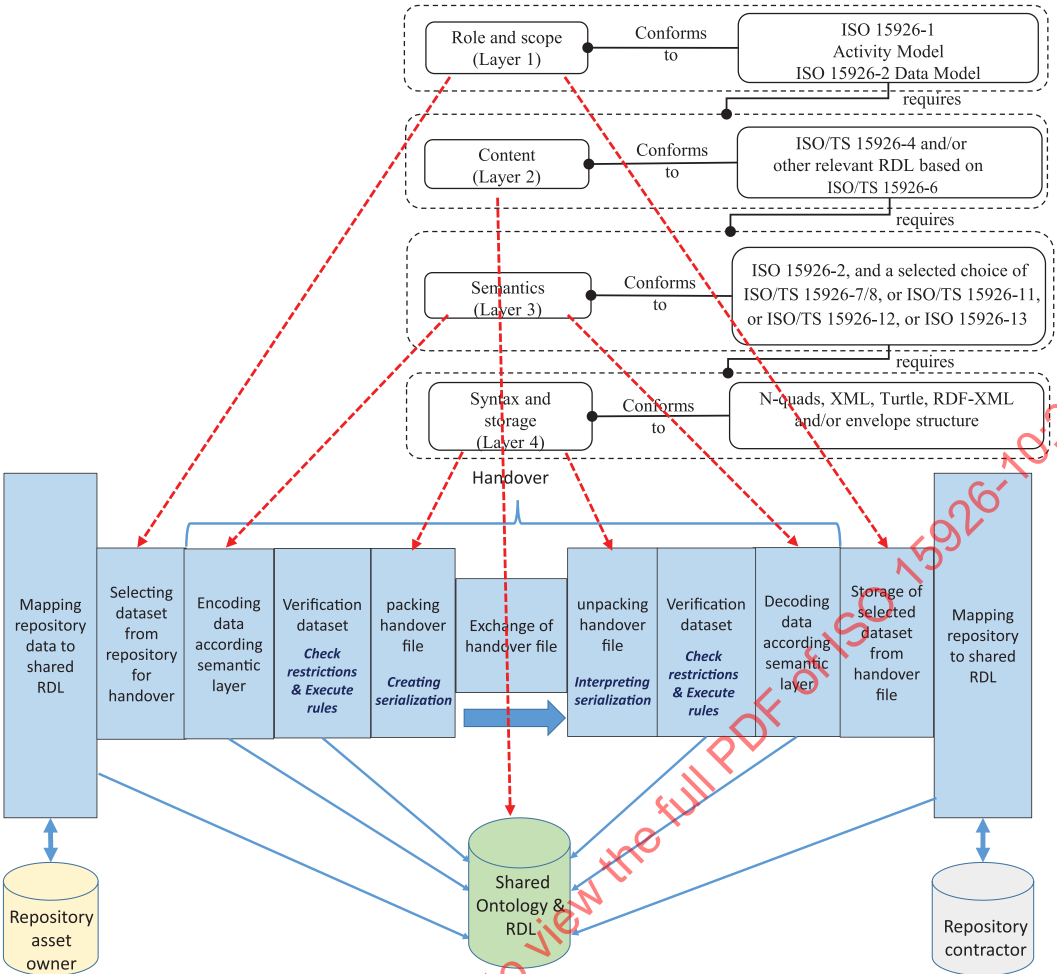

Figure 7 — Layers within the process of data exchange of Figure 3 mapped on the exchange process of Figure 6

Table 2 shows the full relationship between the systematic approaches from the start of stating the business requirement concerning the data exchange layers, passing the realization of the data exchange, up to proof of conformance to these requirements. It is, thus, up to the user requirement captured to identify and declare the exact coverage to the relationship.

Table 2 — Information exchange layer of ISO 15926 and conformance testing

Subject of conformance testing	Predicate of the conformance testing	Object of conformance testing
Business requirements and selecting handover schema.	Validating role and scope (Layer 1).	Quality (usability, completeness, and consistency) of handover schema.
RDL content management requirements and selecting relevant reference data items for handover scheme.	Verifying content (Layer 2).	Availability of the reference data items for the selected handover scheme.
Semantic requirements and selecting appropriate ISO 15926 part(s) for handover.	Verifying semantics (Layer 3).	Correctness of the semantic interpretation of data for handover.
Technological requirements and selecting appropriate serialization and data for handover technology.	Verifying syntax and storage (Layer 4).	Correctness serialization and data for handover technology.

At each layer, specific conformance testing shall be planned and executed, as shown in Table 2. It shall depend on the role and scope defined. The appropriate plan shall be schemed and recorded in the plan.

For each of the scenarios defined in Annexes B to F, the following steps of the conformance testing according to the functional approach as described in ISO/IEC 17000 will be followed (naturally or implicitly):

— selection (of information): planning and preparation activities to collect and produce all information and input needed for the subsequent determination functions;

— determination (gather evidence): developing complete information regarding the fulfilment of the specified requirements by the object of conformity assessment or its sample;

— review (of evidence) and attestation (to conformity): constituting the final stage of checking before taking the critical decision as to whether or not the purpose of conformity assessment has been reliably demonstrated to fulfil the specified requirements.

NOTE In some cases, systematic iteration of the functional approach can be required to maintain the validity of the statement resulting from the review and attestation.

These steps shall be done (more or less) for each of the four data exchange layers for each scenario. This is represented by Figure 8, where via a three-dimensional space, one can address the combination of the layer, step, and scenario. The scenarios follow the structure below:

— overview (of scenario conformance);

— general (of the scenario of the annex and systems involved);

— procedure (steps of the workflow of handover or integration);

— aspects to be validated (verification);

— detailed validation (verification);

— not in scope (define what is not included in the workflow steps but useful to know).

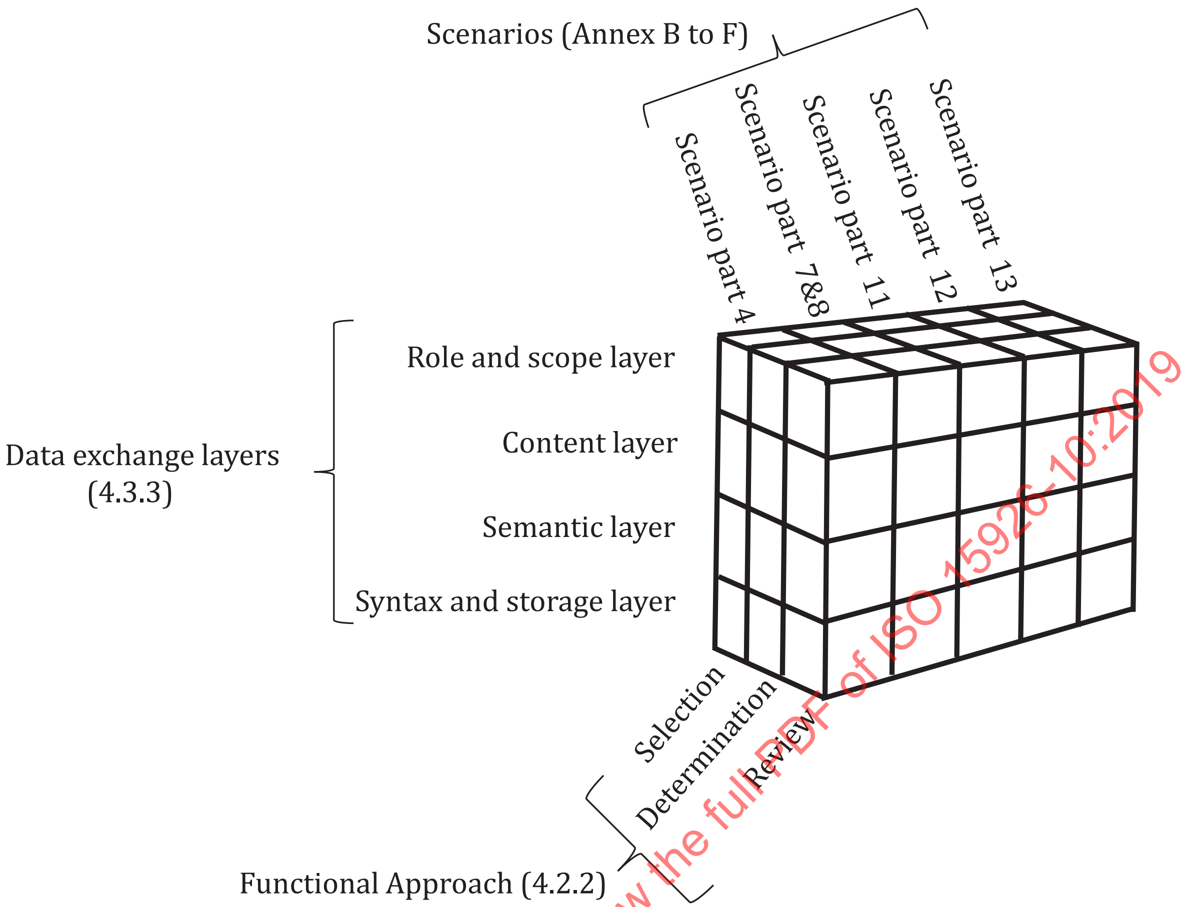

Figure 8 — Combining data exchange layers with conformance steps

4.3.6 Not in scope

Development of software that supports this way of file exchange is outside the scope of this document.

5 Conformance to ISO/TS 15926-7 and ISO/TS 15926-8

5.1 Overview

The data model of ISO 15926-2 is generic and highly normalized. While this enables considerable flexibility in what can be said, it can also give rise to complexity in how it is said.

ISO/TS 15926-7 specifies templates that are expressions of predefined units of semantics, allowing the use of the model in a convenient way.

ISO/TS 15926-8 defines the constraints of the lowered templates of ISO/TS 15926-7 in OWL and implements these lowered templates in RDF.

The data of application programs are mapped from their native format to RDF triples, from there to the signature of TIPs (Template of Information Pattern), and finally transformed to ISO/TS 15926-8 exchange files.

Figure 9 provides an overview of the integration architecture of ISO 15926.

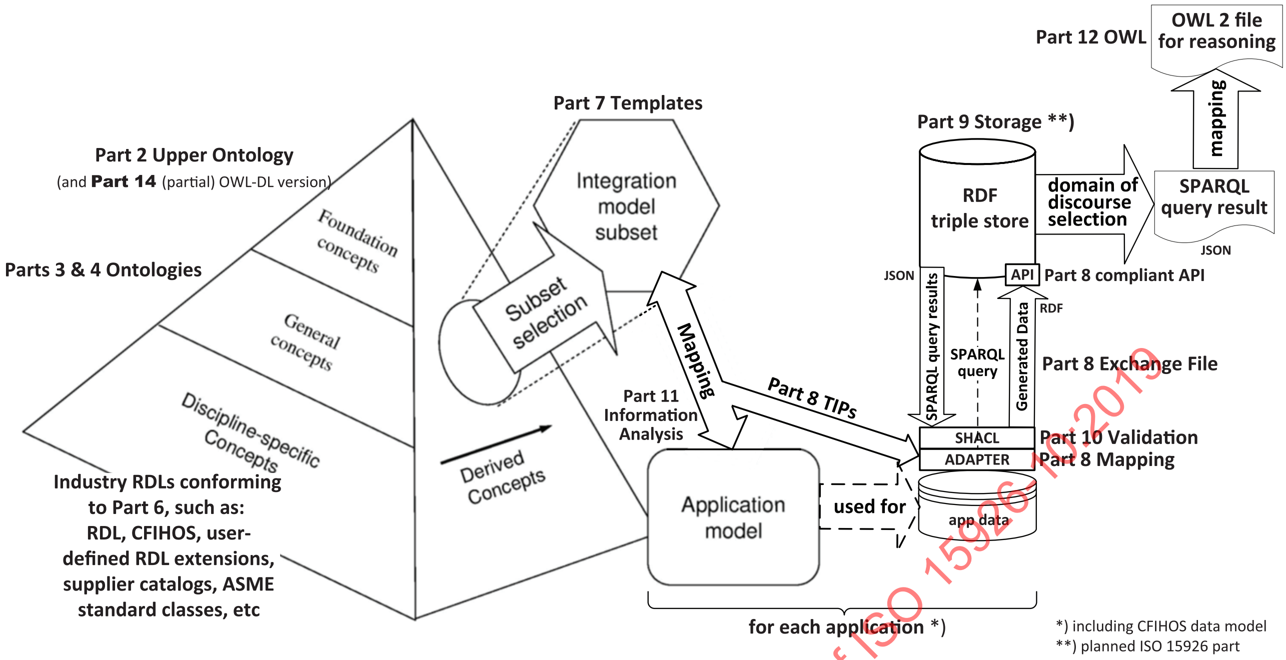

Figure 9 — Overview of the integration architecture of ISO 15926

5.2 General

This clause defines conformance of exchange files to ISO/TS 15926-8.

It defines mapping adapters for COTS (Commercial Of-the-shelve) and home-made software, and exchange files that contain object declarations regarding ISO 15926-2 entity types and instances of ISO/TS 15926-7 Lowered Templates.

These adapters and exchange files can be used for:

— application-to-application interfacing (see Figure 10); or

— information sharing and integration (see Figure 11).

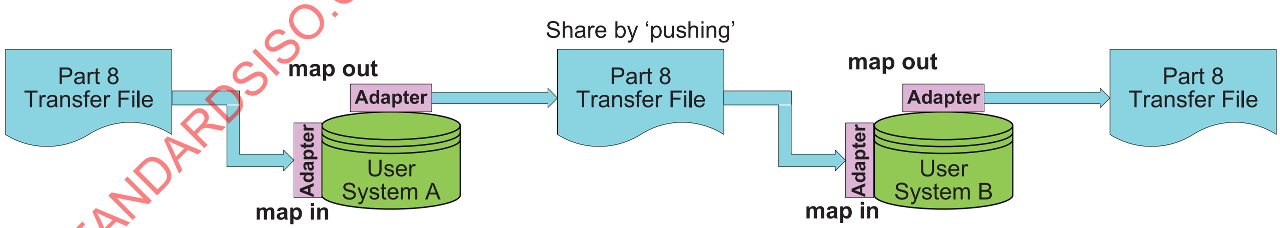

Figure 10 — Application-to-application interfacing

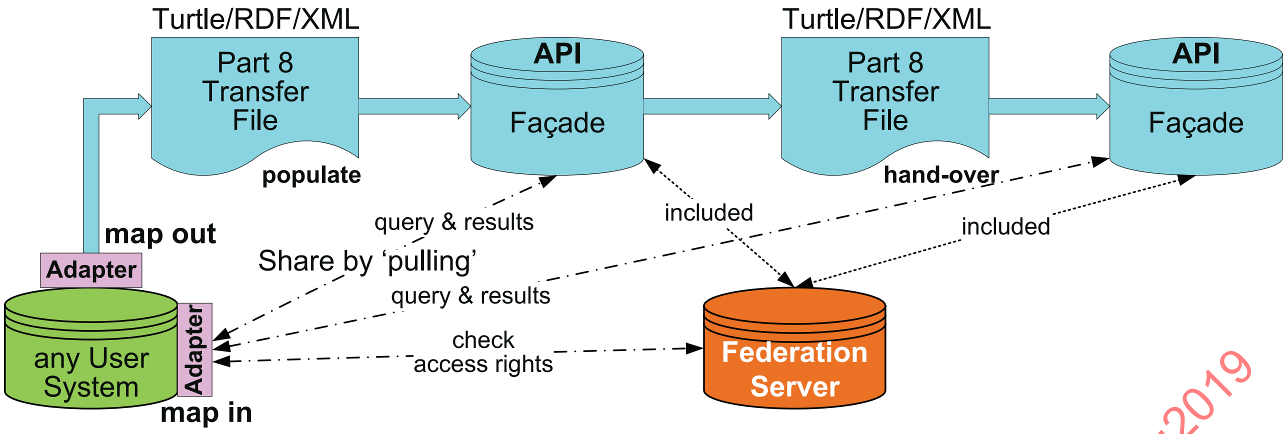

Figure 11 — Information integration

The "map in" adapter is different in both cases because SPARQL queries have a different format (XML, JSON, CSV or TSV) as compared with an ISO/TS 15926-8 Transfer File (Turtle).

NOTE 1 A façade has been defined in ISO/TS 15926-7 as a data warehouse filled with entity instances of entity data types and instances of templates that together provide data integration, and that is used as source and destination for the exchange of lifecycle information.

NOTE 2 Entity data types are defined in ISO/TS 15926-7 as one of the entity data types defined in the data model of ISO 15926-2 and mapped to the XML schema format.

5.3 Procedure steps of the workflow of integration

This document comes at the end of an extensive set of mapping and transforming activities, as illustrated in Figure 12.

When the mapping software has finished its task, the ISO/TS 15926-8 transfer file has been generated and validated; it is ready for uploading to a facade or for sending it to another application. This is the “source-side” validation.

When a SPARQL query result, in the form of an XML, JSON, CSV or TSV file, has been mapped to the internal format of the application under test the semantics of that incoming mapped information can be validated using the presentation capabilities of that application.

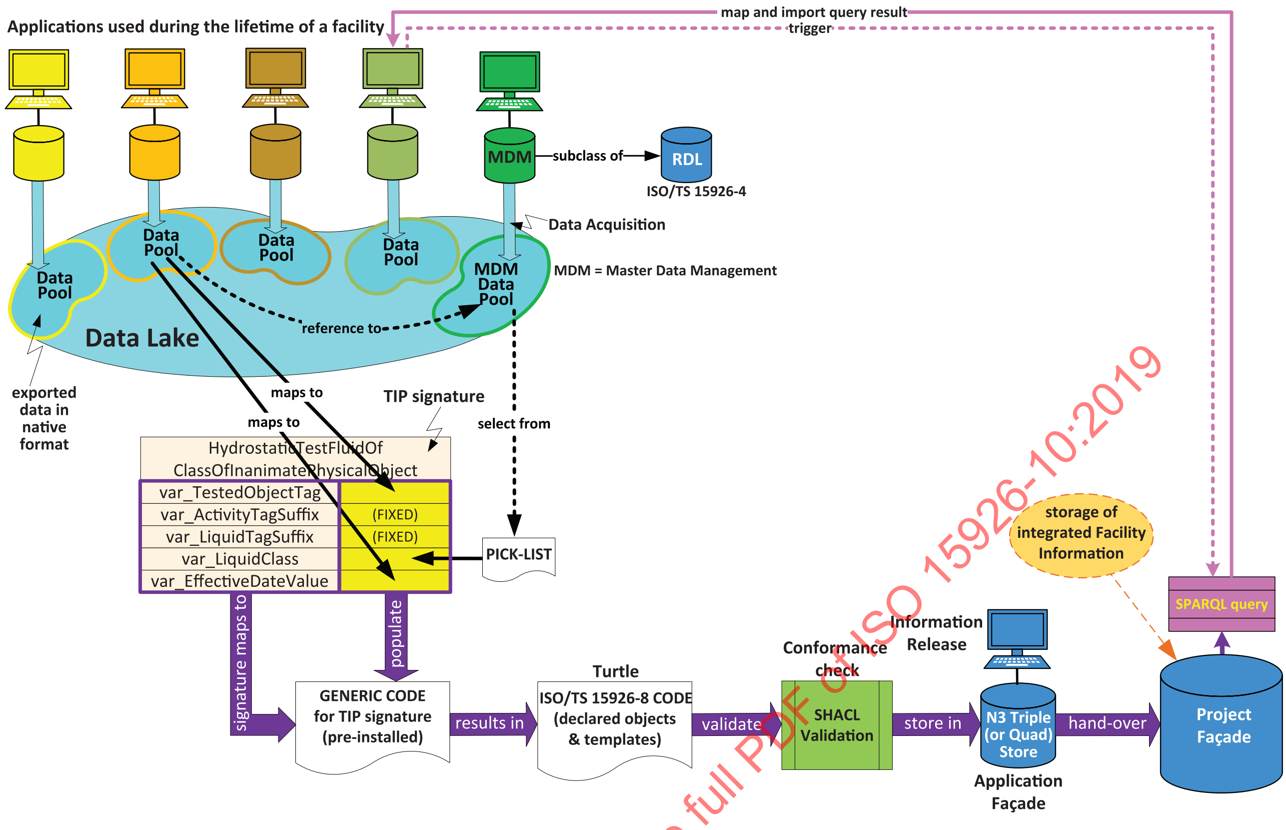

Figure 12 — Overview of activities

5.4 Not in scope

a) The semantical and technical quality of the information before mapping is out of scope.

b) When uploading to a project facade, there is also facade-specific “target-side” validation, which serves to check whether the newly loaded information conflicts with the information that is already stored in the applicable federation of SPARQL endpoints. These “target-side” validations related to the functioning of a project facade are outside the scope of this clause.

6 Conformance to ISO/TS 15926-11

6.1 Overview of scenario conformance

ISO/TS 15926-11 provides the capability to express a product model with RDF triples, RDF Named Graphs and a standardized set of natural language relationships resulting in a table that can be exchanged and shared easily in the industry.

ISO/TS 15926-11 is based on Gellish combined with plain RDF(S), using a specific Named Graph approach.

This Named Graph approach is based on capturing a very limited set of RDF(S) triples within one Named Graph, defining a (main) statement (or “fact”) accompanied by relevant meta data triples.

A triple is expressed as a Subject-Predicate-Object combination where predicates are in accordance with either the following formal RDF(S) ones or the relationships as given in the initial set of relationships as given in ISO/TS 15926-11:

— rdf:type;

— rdf:Property;

— rdfs:Class;

— rdfs:subClassOf;

— rdfs:subPropertyOf.

Figure 13 shows how the initial set of relationships of ISO/TS 15926-11 (with as example part11: consistsOf) and instances of the entity part2:thing, acting as domain and range as well of these relationships, are connected to rdfs:resource.

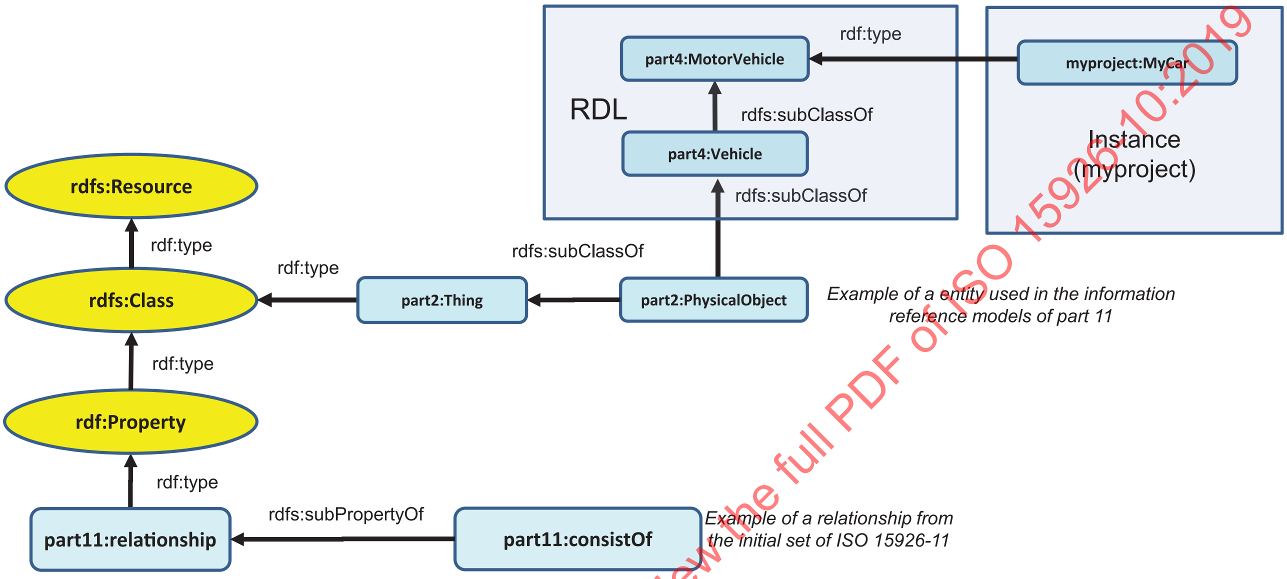

Figure 13 — Connection of ISO/TS 15926-11 with RDF “resource”

Figure 14 shows an example of a system engineering reference information model as defined in ISO/TS 15926-11 consisting of statements which can be exchanged by using so-called “statements graphs” as defined in ISO/TS 15926-11. Individuals are created by means of “individual graph.” Specialized relations are introduced by means of “relationship class graph.” The initial set of relationships which form a normative annex of ISO 15926 are derived from the set of systems Engineering information models as defined in ISO/TS 15926-11.

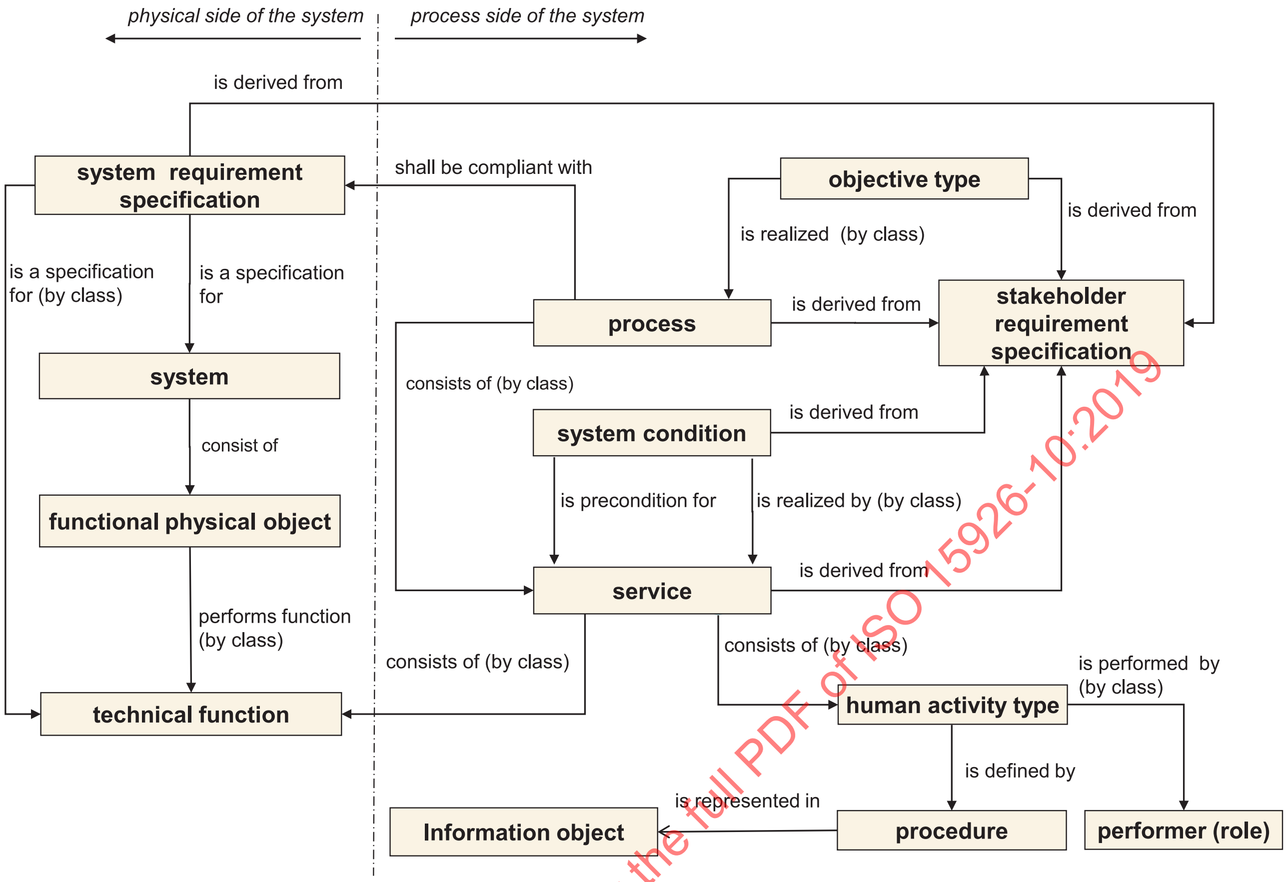

Figure 14 — Example of an information reference model, build with ISO/TS 15926-11 relationships and entities derived from ISO 15926-2

6.2 General

A batch of "SE information" can be captured in a "digital envelope" which will be sent via the WEB portal as shown in Figure 15 (called a facade in ISO 15926). The rules for structuring the content of this digital envelope are expressed in a light core model described by RDF(S) and Named Graph templates. As file and syntax format for the exchange of data, there are several options such as RDF-XML, TriG, Turtle, N-Quad, and even spreadsheets.

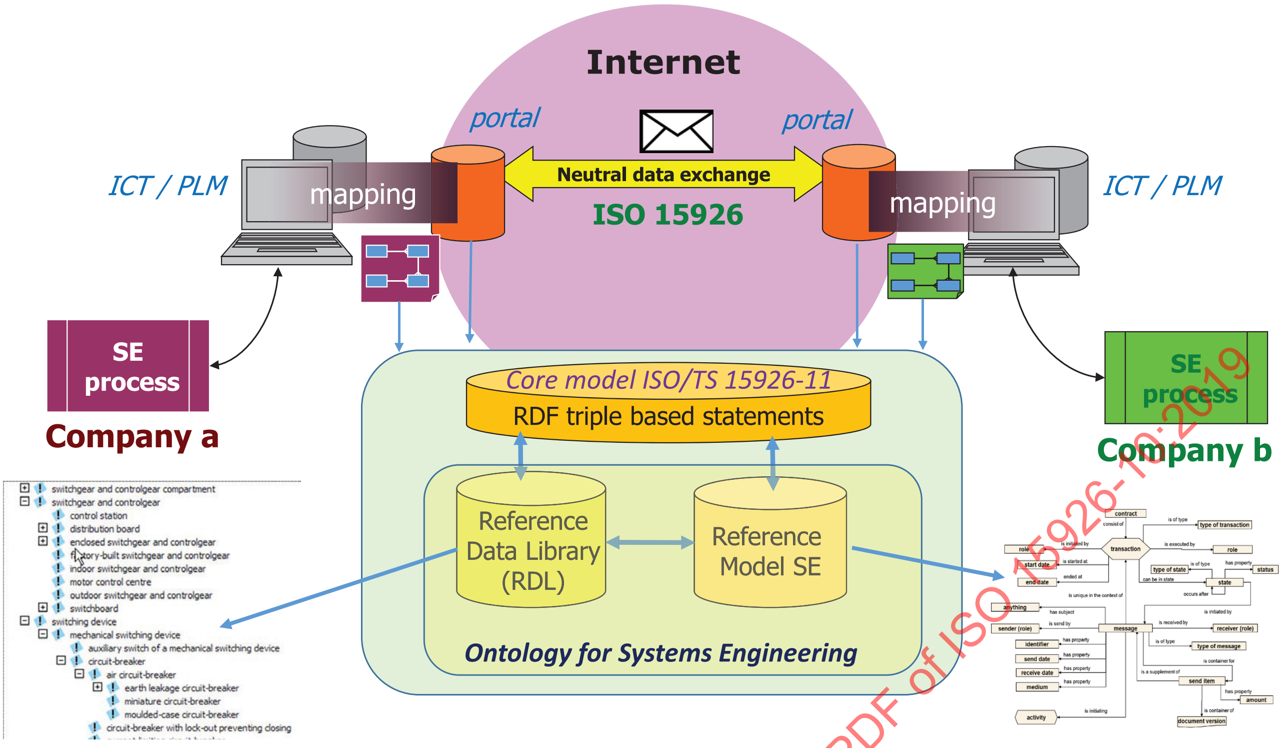

Figure 15 — Information exchange architecture based on ISO/TS 15926-11 using the information models as reference models to validate exchanged data between organizations

NOTE 1 "SE information" means: "Systems Engineering information."

NOTE 2 "SE process" means: "Systems Engineering process."

6.3 Procedure steps of the workflow of handover

Figure 16 shows the validation process of an ISO/TS 15926-11 exchanged file. Starting from an export of e.g. a PLM system of a selected set of project data, the check against the validation rules, means a check if all instances are classified as an element from the specified RDL, if all relationships are in accordance with the information reference model, including prescribed domain and range and cardinality, if the all statement is properly defined using the applicable type of Named Graph in accordance with ISO/TS 15926-11, and if the syntax of the chosen serialization method is correctly used.

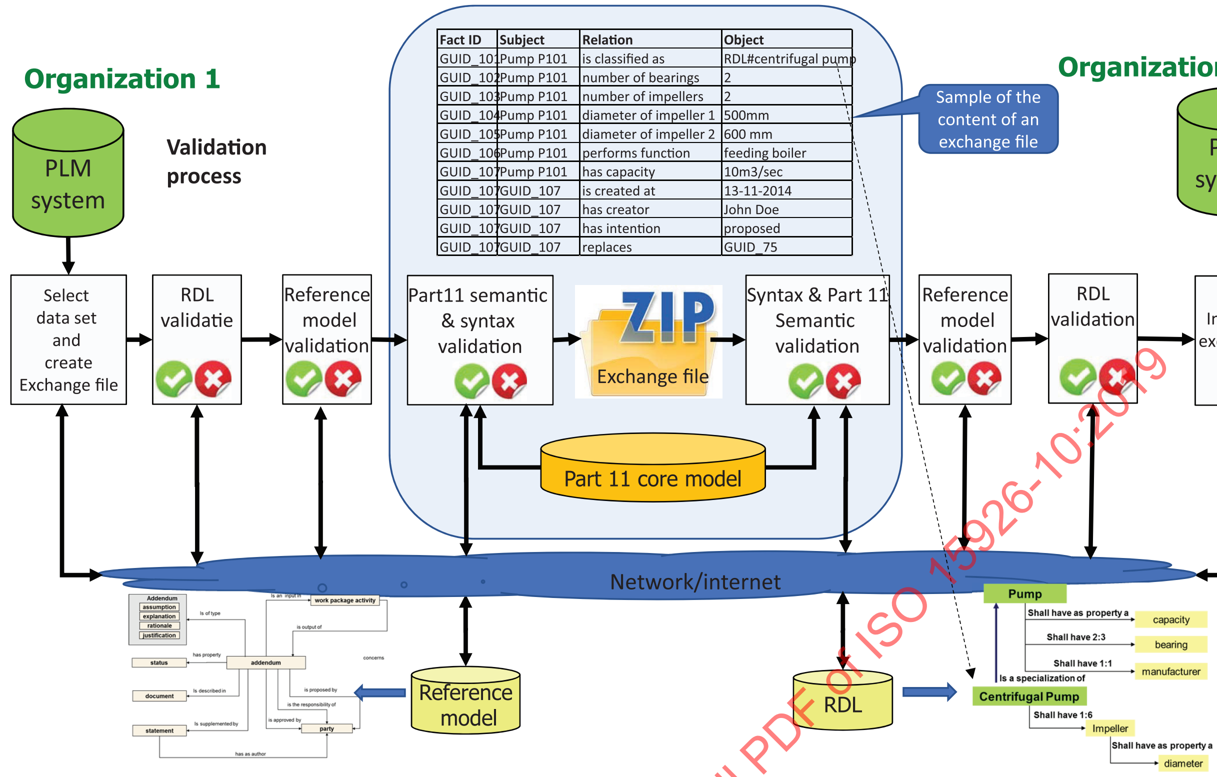

Figure 16 — Validation process of an exchange file in the context of ISO/TS 15926-11

NOTE “PLM system” means “Product Lifecycle Management system.”

6.4 Not in scope

Not in scope is a check of the correctness of the content of the exchange file concerning domain-specific content data.

Annex A
(normative)

Overall conformance to ISO 15926

A.1 Scope of conformance statement

The scope of the conformance statement is restricted to statements about individual things, where:

— individual things are principally physical objects or activities;

— individual things exist or are intended to exist.

This scope supports data about:

— design of actual process plants and actual operational and maintenance activities;

— design of process plants that are intended to exist (including intended future refurbishments of actual process plants);

— plans for intended activities and actual activities that are performed to fulfil the plans.

NOTE 1 ISO 15926 has a much broader scope, but this is a useful subset. The reference data that specifies the types of individuals and the types of their properties can also be held in accordance with ISO 15926. The conformance statement does not deal with the conformance of the reference data, Information objects, and their information content, or identification and description relationships with information objects and their information content.

NOTE 2 An actual individual exists now or has existed in the past. Data recorded about an actual individual is the result of an observation or measurement.

NOTE 3 There are many different approaches to recording intentions. In some of them, an intention is expressed in terms of classes of individual and classes of a possible world. ISO 15926-2 allows different approaches. However, the examples in ISO 15926-2 express an intention as an individual in a single future possible world.

A.2 What is an individual

ISO 15926-2 defines an individual to be one of the following:

— activity: something that happens;

— event: a state change;

— physical object, a quantity of matter;

— person;

— organization;

— region of space;

— period in time,

where a physical object can be:

— functional physical object: a physical object that has functional, rather than material, continuity as its basis for identity [definition in ISO 15926-2];

EXAMPLE 1 The heat exchanger system known as tag E-4507, which is part of a distillate transfer system, is a functional physical object. This is distinct from the "shell and tube heat exchanger manufacture number ES/1234" that was installed as E-4507 when the plant was first built and later removed when worn out, to be replaced by a new heat exchanger with a different serial number [example in ISO 15926-2].

NOTE 1 A functional physical object is commonly identified by a tag.

NOTE 2 Operational data is commonly assigned to a functional physical object. Maintenance data can also be assigned to a functional physical object.

— materialized physical object: a physical object that has matter and/or energy continuity as its basis for identity [definition in ISO 15926-2];

EXAMPLE 2 The shell and tube heat exchanger with manufacture's serial number ES/1234 is a materialized physical object [example in ISO 15926-2].

NOTE 3 A materialized physical object is commonly identified by an asset number or manufacturer's serial number.

NOTE 4 During its life, a materialized physical object can be installed as different functional physical objects at different times. It can therefore play different roles.

— stream: a physical object that is material or energy moving along a path, where the path is the basis of identity [definition in ISO 15926-2].

An activity is a possible individual that brings about change [definition in ISO 15926-2].

EXAMPLE 3 Pumping a fluid with a mechanical pump is an activity [example in ISO 15926-2].

NOTE 5 An activity has individuals that participate in it. A participant in an activity can be a person, organization, or physical object.

NOTE 6 A temporal part of an individual is also an individual. A temporal part of a physical object can be called a "state." A temporal part of an activity can be called a "phase" or "stage." If a property of individual changes with time, then it is assigned to a temporal part.

A.3 Conforming statements about individuals

Conformance for the following statements about individuals is checked as indicated below.

NOTE 1 Statements can be made about individuals that are not in the following itemized list, but conformance for statements not in the list cannot be checked.

— Classification: Each individual shall be classified as a member of one of:

  — activity;

  — event;

  — physical object (or subclass as defined in ISO 15926-2);

  — region of space;

  — period in time.

  NOTE 2 An individual can also be classified as a member of one or more classes that are subclasses of the classes in this list. These classes can be defined in ISO/TS 15926-4 or another reference data library.

— Decomposition: The decomposition of an individual into parts shall be recorded by the ISO 15926-2 relationship composition of individual. No other relationship shall be used for this purpose.

  The composition of individual relationships shall form one or more directed graphs.

NOTE 3 The composition of individual relationship has the specialization arrangement of individual which is used where the decomposition is of an assembly.

NOTE 4 In an implementation, the composition of individual relationship can be renamed. In ISO/TS 15926-12, it is called "has part" with inverse "part of."

— Connectivity: The connection of one individual to another shall be recorded by the ISO 15926-2 relationship connection of individual. No other relationship shall be used for this purpose.

NOTE 5 In an implementation, the connection of individual relationship can be renamed. In ISO/TS 15926-12, it is called "connected to."

— Properties: The physical properties of an individual shall be recorded by the ISO 15926-2 relationship indirect property. No other relationship shall be used for this purpose.

An indirect property shall be classified to specify the nature of the property.

EXAMPLE The pump with tag "P-101" has a "maximum operating pressure" of 10 bar. The indirect property relationship between "P-101" and 10 bar is therefore classified as "maximum operating pressure."

NOTE 6 In an implementation, the indirect property relationship can be renamed and implemented differently. In ISO/TS 15926-12, it is called "physical property."

NOTE 7 In the ISO/TS 15926-12 implementation of example 1, "maximum operating pressure" is an owl: FunctionalProperty which is classified as a "physical property."

— Participation: The participation of an individual in an activity shall be recorded by the ISO 15926-2 relationship participation. No other relationship shall be used for this purpose.

NOTE 8 A participant in an activity is usually a temporal part of an individual that exists before the activity begins and continues to exist after the activity has ended.

A.4 Changes to individuals

Change to an individual can arise as follows:

— individual can change through time as a result of an activity;

NOTE 1 An activity can be carried out by people or a physical process.

— intended individual can be changed by a planning or design decision.

NOTE 2 The reason for the decision can be a change to the requirement or a change to the proposed technical solution.

NOTE 3 ISO 15926-2 states that a statement made about an individual is true for the entire life of the individual. The classification of an individual therefore applies to the individual for its entire life; the individual has the same connection and composition relationships for its entire life; the individual has the same properties for its entire life; the individual that participates in an activity does so for its entire life.

Change to an individual shall be recorded as follows:

— Change over time as a result of activity: This shall be recorded by statements that are about temporal parts of an individual.

The whole for a temporal part shall be recorded by a temporal whole part relationship.

NOTE 4 A temporal part is itself an individual and has a beginning and an end time. A statement about a temporal part is true for the duration of the temporal part.

EXAMPLE 1 The pump with tag "P-101" is running from 10:00 to 11:00 on 2018-08-10 and not running from 11:00 to 12:00. This is recorded by defining two temporal parts, or "states," of pump "P-101" (one from 10:00 to 11:00 on 2018-08-10 and one from 11:00 to 12:00); classifying the former as "running" and the latter as "not running."

— Change as a result of a planning or design decision: This shall be recorded by statements that are about different intended individuals. If an intention changes, then it is a different individual that is intended.

EXAMPLE 2 The intended pump with tag “P-101” initially has a rated power of 1 kW. On 2018-08-10, a design decision increases the rated power to 1,5 kW. This is recorded by defining two individuals “P-101 v-1” and “P-101 v-2”; assigning a “rated power” property of 1 kW to the former and 1,5 kW to the latter.

A.5 View models for ISO 15926

The approach to change defined by ISO 15926-2 is not suitable for all implementations. It is permitted for an ISO 15926 implementation to take a different approach provided that data held in accordance with the implementation can be unambiguously translated into data that conforms to ISO 15926-2.

EXAMPLE 1 The implementations defined by ISO/TS 15926-7 and ISO/TS 15926-8, and by ISO/TS 15926-11 both assign effectivities to statements. Their approaches are different: ISO/TS 15926-7 and ISO/TS 15926-8 have reified statements for which beginning and end times can be assigned; ISO/TS 15926-11 uses RDF Named Graphs to assign metadata to collections of statements.

In either case, the approach can be used to assign a statement to a particular temporal part of an individual or a particular version of a design.

This approach is not inconsistent with ISO 15926-2 because data held in accordance with the implementation can be unambiguously translated into data that conforms to ISO 15926-2.

EXAMPLE 2 ISO 15926-13 defines a view model which provides a simplified approach to planning data. In this view model, an activity object is both the planned activity and the actual activity. This object has a planned start date; an actual start date; a planned finish date; an actual finish date.

When translated into data that conforms to ISO 15926-2, there would be two individuals: the planned activity and the actual activity. Each would have a single start date and a single finish date.

Annex B
(normative)

Conformance to ISO/TS 15926-4

B.1 General

ISO/TS 15926-4 provides a basic level of engineering understanding that can be used by very many systems for the creation and management of process plant data. A reference to ISO/TS 15926-4 shall specify the nature of an object or property in engineering terms.

NOTE 1 Two systems that both conform to ISO/TS 15926-4 thereby share a basic engineering understanding.

NOTE 2 Many systems for the creation and management of process plant data do not have overall conformance to ISO 15926, as described in Annex A. The difference between the structure of the data held by a system and ISO 15926 can be so great that to map between the two is excessively complicated.

B.2 Conformance to ISO/TS 15926-4

B.2.1 Scope of conformance statement

The scope of the conformance statement concerns conformance of an implementation to ISO/TS 15926-4, without necessarily having overall conformance to ISO 15926.

NOTE 1 Conformance of implementation is assessed by criteria applied to an external view of data created by an application. An external view can be provided by an exchange file or by an API. Conformance is not assessed by criteria applied to internal data stored by an application.

ISO/TS 15926-4 contains a reference data library that can be used to classify objects in engineering and asset management data. This conformance statement is limited to the classification of the following:

— physical objects and activities;

— properties (also called “quantities” in ISO/TS 15926-12);

— indirect properties (also called “physical properties” in ISO/TS 15926-12);

— property quantification, by reference to a scale or unit of measure.

NOTE 2 Most of the content of ISO/TS 15926-4 is concerned with subclasses or subproperties of these classes and properties.

B.2.2 Purpose of conformance

In an implementation conforms to ISO/TS 15926-4, the nature of a physical object, activity, property or scale in data created by the implementation shall be defined by reference to the reference data library of ISO/TS 15926-4.

NOTE 1 This means that a receiving system can rely upon the reference data library of ISO/TS 15926-4 in order to interpret the data.

NOTE 2 For a physical object, activity or property, the reference is made by a classification relationship. In an RDF/OWL implementation, this relationship is rdf:type. For an indirect property or scale, the way that the reference is made depends upon the implementation approach, as follows:

— implementation, such as that defined by ISO/TS 15926-8, can regard an indirect property or property quantification as an object. In this case, the reference to ISO/TS 15926-4 is made by a classification relationship;

— implementation, such as that defined by ISO/TS 15926-12, can regard an indirect property or property quantification as an RDF statement. In this case, the subproperty defined in ISO/TS 15926-4 is referenced as the predicate.

B.2.3 Criteria for conformance

Conformance of implementation to ISO/TS 15926-4 has two levels as follows:

— Strict: If it is necessary to specify the nature of a physical object, activity, property, indirect property or property quantification, then an implementation shall use a reference data item defined in ISO/TS 15926-4, or a reference data item that is stated to be a specialization of a reference data item defined in ISO/TS 15926-4.

  NOTE 1 It is always necessary to classify an indirect property or property quantification because the nature of these relationships cannot be inferred from other data.

  NOTE 2 The nature of a property can often be inferred from the nature of an indirect property or property quantification that references it.

  NOTE 3 A reference to a reference data item in ISO/TS 15926-4 can be made by use of its URI, unique number, or unique name.

  EXAMPLE If an implementation that conforms to ISO/TS 15926-4 defines a physical object that is a pump, then it shall be classified as a member of the class "pump" defined in ISO/TS 15926-4, or by a user-defined class that is stated to be a subclass of pump.

— Partial: Partial conformance to ISO/TS 15926-4 relaxes the requirement that a reference data item shall be defined in ISO/TS 15926-4, or stated to be a specialization of an item defined in ISO/TS 15926 -4. Instead, a reference data item that is not a specialization of an item defined in ISO/TS 15926 -4 is permitted.

  However, a reference data item that is a duplicate of a reference data item in ISO/TS 15926-4 shall not be used, and where a reference data item is a specialization of a reference data item in ISO/TS 15926-4, this shall be stated.

  NOTE 4 Partial conformance to ISO/TS 15926-4 is relevant to an application area that is only partly within the scope of ISO/TS 15926-4.

Annex C
(normative)

Conformance to ISO/TS 15926-7 and ISO/TS 15926-8

C.1 Aspects to be validated

Several aspects shall be validated. The list below tries to be exhaustive:

— is the ISO/TS 15926-8 file well-formed Turtle or RDF/XML;

— has the information been properly mapped from a semantics point of view;

— has each Thing, that is referred to, by any template signature, been declared;

— have all declarations been placed in the correct life cycle activity;

— have all declarations been using the correct ISO 15926-2 entity type;

— have all declarations been classified/specialized with/from the correct instance of "ClassOfFunctionalObject";

— are the template role fillers of the prescribed ISO 15926-2 entity type or a subtype thereof;

— are the endpoints of the role fillers part of the applicable endpoint federation.

NOTE Any outgoing ISO/TS 15926-8 exchange file can be validated by the rules outlined above.

C.2 Detailed validation

C.2.1 Well-formed RDF

Validate well-formedness of any:

— Turtle file with a Turtle Validator, comparable with http://rdf-translator.appspot.com/;

— RDF/XML file with an RDF Validator, comparable with http://www.w3.org/RDF/Validator/ .

C.2.2 Semantics quality

The testing authority shall set up facade with a browser that displays the results of any incoming ISO/TS 15926-8 file. It shall also store the information that the system under test shall be able to fetch with a SPARQL query and map the results, therefore to its internal format.

C.2.3 Declarations

Validate that all rdf:objects have been declared in accordance with ISO/TS 15926-8:2011, C.8.1.

EXAMPLE 1 a Class declaration:

```
:10bba3a5-e0b4-49b1-bf95-c431a243ba9b rdf:type dm:ClassOfInanimatePhysicalObject;
  rdfs:subClassOf rdl:RDS414674;
  rdfs:label "CO_B14-V-101";
  meta:hasLifecycleActivity rdl:RDS222415 ; # VESSEL ENGINEERING
  meta:valEffectiveDate "2015-06-24T14:57:00Z"^^xsd:dateTime .
```


EXAMPLE 2 a PossibleIndividual declaration:
:b572e2a7-2c35-48bf-87d7-bbac6a1d65e9 rdf:type
dm:FunctionalPhysicalObject, dm:WholeLifeIndividual, lci:NonActualIndividual, lci:InanimatePhysicalObject, rdl:RDS414674;
  rdfs:label "B14-V-101";
  meta:hasLifecycleActivity rdl:RDS222415;
  meta:valEffectiveDate "2015-10-14T13:00:00Z"^^xsd:dateTime.

EXAMPLE 3 a template declaration:
:ffbb44bb-fb29-4a53-8a8d-17f9a33b2ca1 rdf:type tpl:ClassOfIndividualHasIndirectPropertyWithValue
;
  tpl:hasPossessorType :cccd040c-b212-4ee2-aa7e-01d5c9fafe4a;
  tpl:hasIndirectPropertyType rdl:RDS357119 ; # OPERATING TEMPERATURE;
  tpl:valPropertyValue "20"^^xsd:decimal;
  tpl:hasScale rdl:RDS1322684 ; # DEGREE CELSIUS;
  meta:hasLifecycleActivity rdl:RDS9701927 ; # PROCESS ENGINEERING;
  meta:valEffectiveDate "2014-06-22T00:00:00Z"^^xsd:dateTime.

C.2.4 Entity type of role fillers

For each role of a template, the role fillers shall be of the ISO 15926-2 entity type or a subtype thereof, as defined in the applicable Template Specification.

EXAMPLE rdl:RDS357119 in the above template is an instance of dm:ClassOfIndirectProperty, and this is as defined in the Template Specification of tpl:ClassOfIndividualHasIndirectPropertyWithValue.

C.2.5 Conformance with ontology

Any template for a PossibleIndividual shall be validated against the applicable ontology, if available. This information comes in the form of templates for Classes.

NOTE At the time of publication, the RDL does not yet contain an ontology definition. At the level of project data, such ontologies can be present.

C.2.6 Endpoints in federation

All facades on a project are federated. This federation shall be queried to verify that all endpoints of all role fillers in the templates that are part of the exchange file are indeed in that federation.

NOTE Each ISO/TS 15926-8 exchange file starts with a list of prefixes that can compare with the definition of the federation before the actual validations begin, although not all those prefixes are necessarily used in the exchange file.

C.2.7 Validation standard

For the validation of the mapping results, the W3C Recommendation “Shapes Constraint Language (SHACL)” shall be used.

This is possible because an ISO/TS 15926-8 exchange file has two types of constructs:

— declared objects;

— template instances.

These are generated using some generic TIPs (Templates for Information Patterns), and for each TIP, a SHACL “shape” is defined, that defines the rules for validation of the mapping results.

NOTE SHACL is a language for validating RDF graphs against a set of conditions. These conditions are provided as "shapes" and other constructs expressed in the form of an RDF graph. RDF graphs that are used in this manner are called "shapes graphs" in SHACL, and the RDF graphs that are validated against a shapes graph are called "data graphs." As SHACL shape graphs are used to validate that data graphs satisfy a set of conditions they can also be viewed as a description of the data graphs that do satisfy these conditions.

Annex D
(normative)

Conformance to ISO/TS 15926-11

D.1 Aspects to be validated

Fundamental aspects of ISO/TS 15926-11 to be validated concerns:

— correct use of the relationships as defined in the initial set of ISO/TS 15926-11

EXAMPLE Correct name and correct domain and range.

— correct specialization of relationships as defined in the initial set;

— correct use of the reference systems engineering, information model;

— correct use of the three types of Named Graphs;

— correct use of meta data within Named Graphs;

— correct use of RDF(S) as defined in ISO/TS 15926-11;

— every instance of an ISO 15926 class has a reference to an RDL element;

— correct use of the chosen serialization syntax (like RDF-XML, Turtle, TRiX).

D.2 Detailed validation

Relationships in the context of making statements about Systems Engineering data (as can be found as reference information models, defined in ISO/TS 15926-11) can be specialized by means of narrowing the domain and or range of an ISO 15926 part-relationship (by using subclasses of the original range and domain classes).

When statements are exchanged, only relationships in the prescribed direction shall be used. The inverse name of the relationship shall only be used in end-user application combined by changing the subject and object from the position in the statement.

Every statement has its RDF Named Graph where every Named Graph has at least as metadata the creator and date/time of its creation.

In ISO/TS 15926-11 allowed statements about statements (meta data) are recognizable in exchanged data, and concerns:

— creator of a statement (party or role);

— date and time of the creation of the statement;

— modifier of a statement (party or role);

— date and time of modification of a statement;

— certainty of a statement; possible values (instances) for certainty can be “estimated,” “calculated” or “as-built”, supporting of ranking the probability of the correct value of property values;

— modality of a statement; possible modal verbs to express the modus of a statement: possible values (instances) of modality are “Can be,” “Shall be” and “Shall have” supporting requirement management and modelling product knowledge. Default the modus of a statement will be “is the case”;

— intention of a statement, possible values (instances) of intention is: “requested,” “proposed,” “approved.” Supporting the workflow in the exchange process and change management, especially for property values;

— cardinality of an element (specific the “object playing role 2”) within a statement, supporting requirement management and product knowledge modelling;

— versioning of a statement, supporting library versioning and baselining in the context of configuration management;

— relating a statement to a specific system life cycle, making the difference between information that is relevant to the conceptual design, detailed design or the maintenance stage of a system;

— defining the “begin of life” and “end of life” of a statement to be able to pinpoint the period in time that the specific statement is valid.

Names of subjects, predicates, and objects are URI’s, which are global in scope, always referring to the same entity in any RDF document in which they appear; objects can also be given as text values, called literal values, which are sometimes typed using XML Schema data types. In ISO/TS 15926-11 URIs are as follows:

— URI (Uniform Resource Identifier) is a compact string of characters that is used to identify a subject, predicate, object or Named Graph;

— hash (#) URI is used whenever one wants to refer to something that doesn’t live on the web, with the base URI providing information about that thing;

— slash (/) URI is used whenever one wants to refer to something live and addressable on the web;

— namespace is a container for a set of identifiers (names). Namespaces usually group names based on their functionality. A namespace can be part of a URI;

— absolute URI reference consists of three parts: a scheme, a scheme-specific part, and a fragment identifier.

The use shall be made of three types of “building constructs” or also called Named Graph templates used when implementing the ISO/TS 15926-11 methodology, all three represented by a specific type of Named Graph (see Figure D.1):

— “individual graph” whereby any new object class or class instance will be introduced in the product, project and or process model. An “individual graph” is only a placeholder for that specific class or instance, and adds no semantics to that class or instance at this point, only defines a label for the URI of the object class or class instance (see Figure D.2);

— “relationship class Graph.” This kind of graph introduces, classifies and prescribes a relationship in terms of the name and allowed domain and range (see Figure D.3);

— “statement graph” which is used to make a concrete statement about another statement or any object introduced in an ”individual graph” (see Figure D.4).

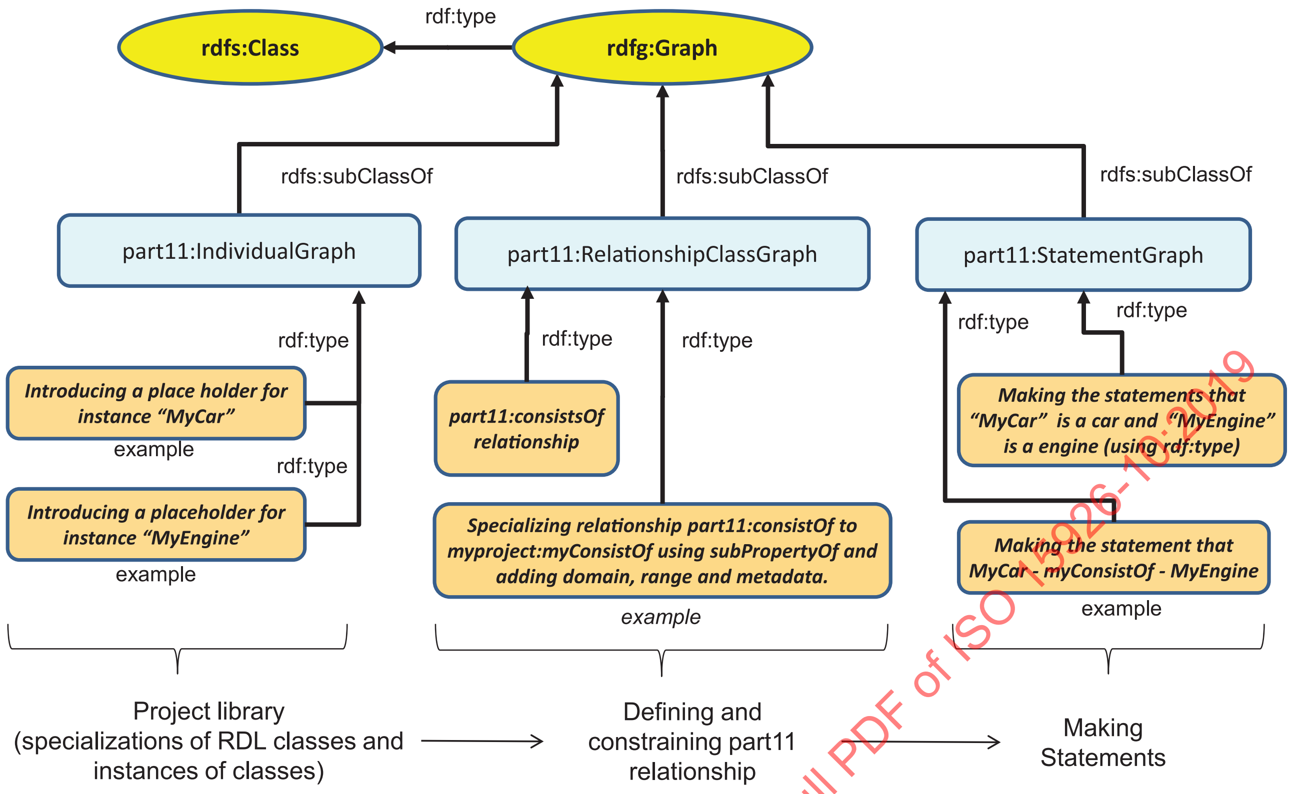

Figure D.1 — Three kinds of Named Graphs: individual graph, relationship graph, and statement graph

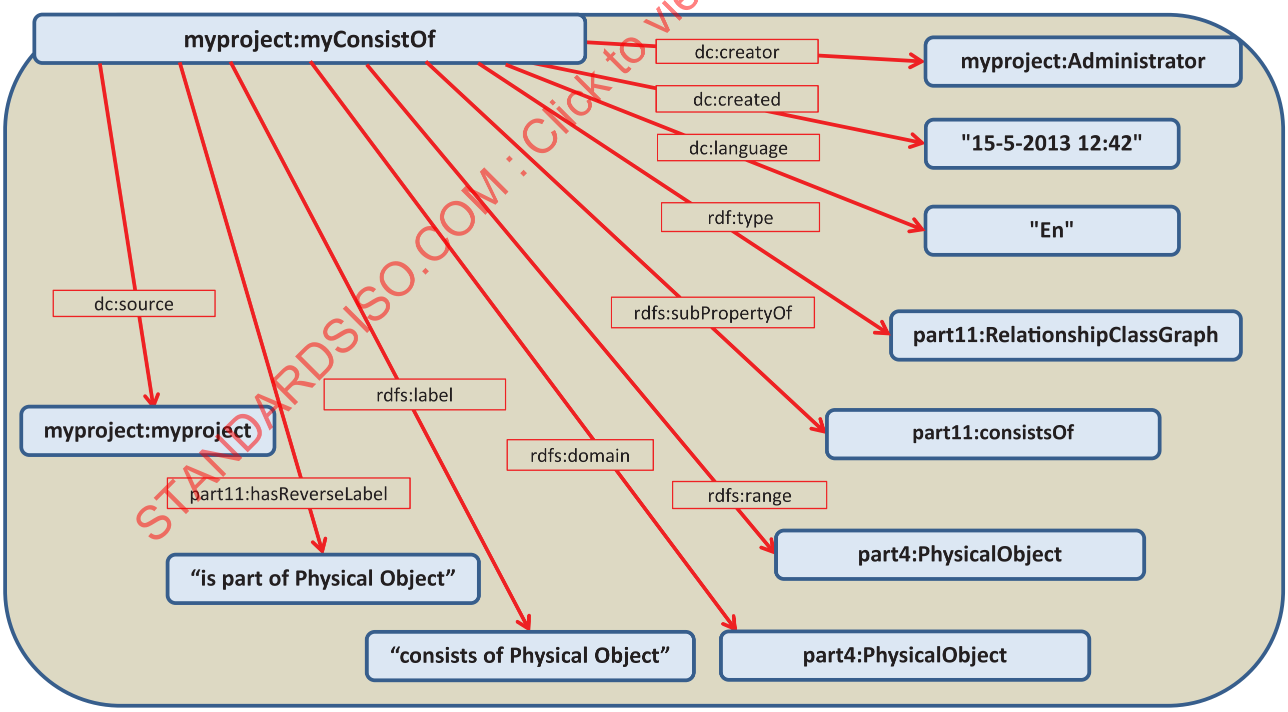

Figure D.2 — Example of a relationship graph: RDF Named Graph introducing a specialized relationship from part11:consistsOf

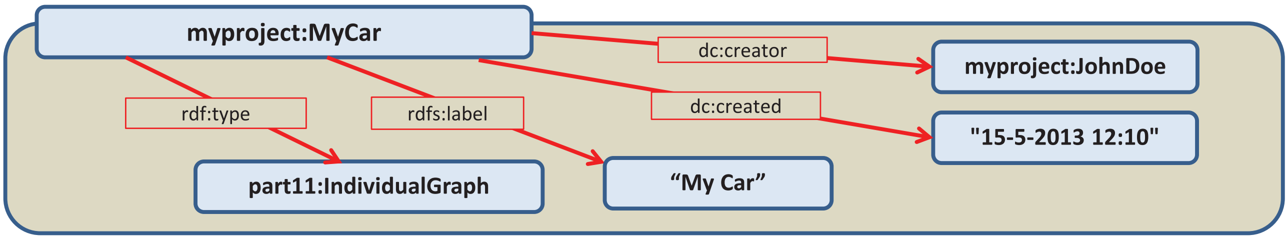

Figure D.3 — Example of an individual graph: RDF Named Graph introducing a placeholder with the label "MyCar"

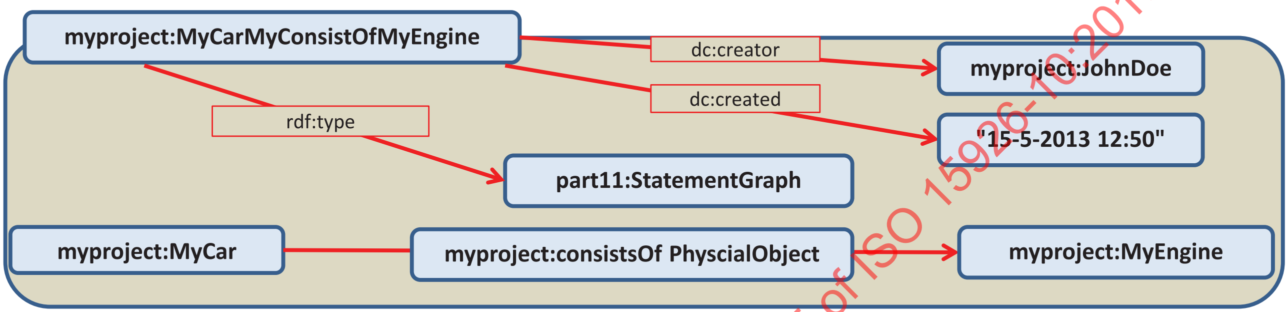

Figure D.4 — Example of a statement graph: RDF Named Graph stating that "MyCar consistsOfPhysicalObject of MyEngine"

ISO/TS 15926-11 defines two scenarios on how to manage changes in information represented by Named Graphs:

— modifications of Named Graphs are allowed: a modified Named Graph then gets as extra metadata: the date/time of modification, additional to the date/time of creation and creator;

— no modifications of any graph are allowed to support full traceability of history (nothing will be ever deleted). The specific graph, subject of the modification, will be replaced by a new one, with a "replaces" relationship to the URI of the superseded graph. This "replaces" relationship can be contained in the same Named Graph as the new Named Graph that supersedes the old Named Graph or can be a "is replaced by" relationship in the old Named Graph or can be contained in a separate Named Graph using a "is replaced by" relationship. Depending on the context of the application, one can choose to implement one of these three mechanisms.

In order to position a Named Graph in this context, meta data can be added. For this purpose, the following terms from the Dublin Core Metadata Initiative are adopted within the ISO/TS 15926-11 methodology:

— dc:creator;

— dc:created;

— dc:modified;

— dc:source;

— dc:language;

— dc:description;

— dc:provenance;

— dc:replaces, dc:isReplacedBy.

From the RDFS namespace the following predicates are used to describe meta data of Named Graph:

— rdfs:label;

— rdfs:domain;

— rdfs:range.

In addition, the following relationships are defined in the initial set of relationships in order to define meta data defining context of a Named Graph (written in camel case starting with a lowercase):

— part11:hasReverseLabel;

— part11:domainIsRestrictedByClass;

— part11:rangeIsRestrictedByClass;

— part11:hasModality;

— part11:hasIntention;

— part11:hasCertainty;

— part11:hasCardinality;

— part11:hasMinCardinality;

— part11:hasMaxCardinality;

— part11:hasAsBeginOfLife;

— part11:hasAsEndOfLife;

— part11:isModifiedBy;

— part11:concernsStage.

The general semantic relationships that are used should be originating from the initial set of relationships or are a specialization of a relationship from the initial set.

For each relationship within the initial set is defined what the highest upward class in the hierarchy is for the subject and for the object of a relationship. Each subclass of the defined highest upward class in the hierarchy is thus allowed as the subject (domain) respectively the object (range) of a relationship.

When one wants to restrict the usage of a relationship in terms of domain and range, a specialization of a specific relationship can be created using the relationship class Graph.