INTERNATIONAL STANDARD

ISO/IEC 15474-2

First edition
2002-10-15

Information technology — CDIF framework —

Part 2: Modelling and extensibility

Technologies de l'information — Cadre de référence CDIF —

Partie 2: Modélisation et extension

Reference number
ISO/IEC 15474-2:2002(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Table of Illustrations

Table of Tables

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 15474 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 15474-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and system engineering*.

ISO/IEC 15474 consists of the following parts, under the general title *Information technology — CDIF framework*:

— *Part 1: Overview*

— *Part 2: Modelling and extensibility*

Introduction

This standard will assist the vendors and users of modelling tools and meta-data repositories in developing mechanisms for interchanging information. This standard specifies an element of a family of related standards. When used together, these standards specify a mechanism for transferring information between tools.

ISO/IEC 15474-1:2002, *Information technology — CDIF framework — Part 1: Overview* should be read first when initially exploring CDIF. It explains the overall CDIF Architecture and how the family of standards fits together.

This standard**,** ISO/IEC 15474-2:2002, *Information technology — CDIF framework — Part 2: Modelling and extensibility*, should also be read before the other standards in the CDIF Family of Standards**.** It defines the CDIF meta-metamodel and the modelling concepts used throughout CDIF and the extensibility mechanism.

This standard has been developed with the wide support and participation of vendors, users, academia and government involved in or familiar with the CASE industry, its products and the general requirements associated with interchanging information between these products.

This document is organized into the following Clauses:

— Clause 1 to 5 are prescribed ISO/IEC Clauses

— Clause 6: Meta-metamodel concepts and facilities

This defines the meta-metamodel, modelling rules and graphical conventions that are used. The data types supported in metamodels are introduced. The concept of the working metamodel is also explained. It also describes the types of extensibility that may be used when extending the CDIF semantic metamodel.

This Clause also describes the responsibilities of an exporter tool with regard to overlapping semantics and maximum output, and those of an importer tool with regard to the retention of information.

— Clause 7: CDIF modelling conventions

This contains the diagramming and documentation conventions used in the CDIF semantic metamodels and in the meta-metamodel specified in this document.

— Clause 8: Meta-metamodel overview

This contains an overview of the CDIF meta-metamodel. This clause includes a meta-metamodel diagram.

— Clause 9: Meta-metamodel summary

This contains an overview of the CDIF meta-metamodel. This clause includes an *AttributableMetaObject* hierarchy, a meta-meta-entity summary and a meta-meta-relationship summary

— Clause 10: Meta-metamodel specification

This contains the full definitions of all the objects in the CDIF meta-metamodel.

Information technology — CDIF framework —

Part 2:
Modelling and extensibility

1 Scope

The CDIF family of standards is primarily designed to be used as a description of a mechanism for transferring information between modelling tools. It facilitates a successful transfer when the authors of the importing and exporting tools have nothing in common except an agreement to conform to CDIF.

The CDIF family of standards includes a semantic metamodel and a transfer format definition. It also includes the specification of a meta-metamodel and associated rules that define a framework for the semantic metamodel and the transfer format. The language that is defined for the transfer format also has applicability as a general language for Import/Export for repositories. The CDIF semantic metamodel also has applicability as the basis of standard definitions for use in repositories.

The standards that form the complete family of CDIF Standards are documented in ISO/IEC 15474-1:2002, *Information technology - CDIF framework - Part 1: Overview*. These standards cover the overall framework, the transfer format and the CDIF semantic metamodel.

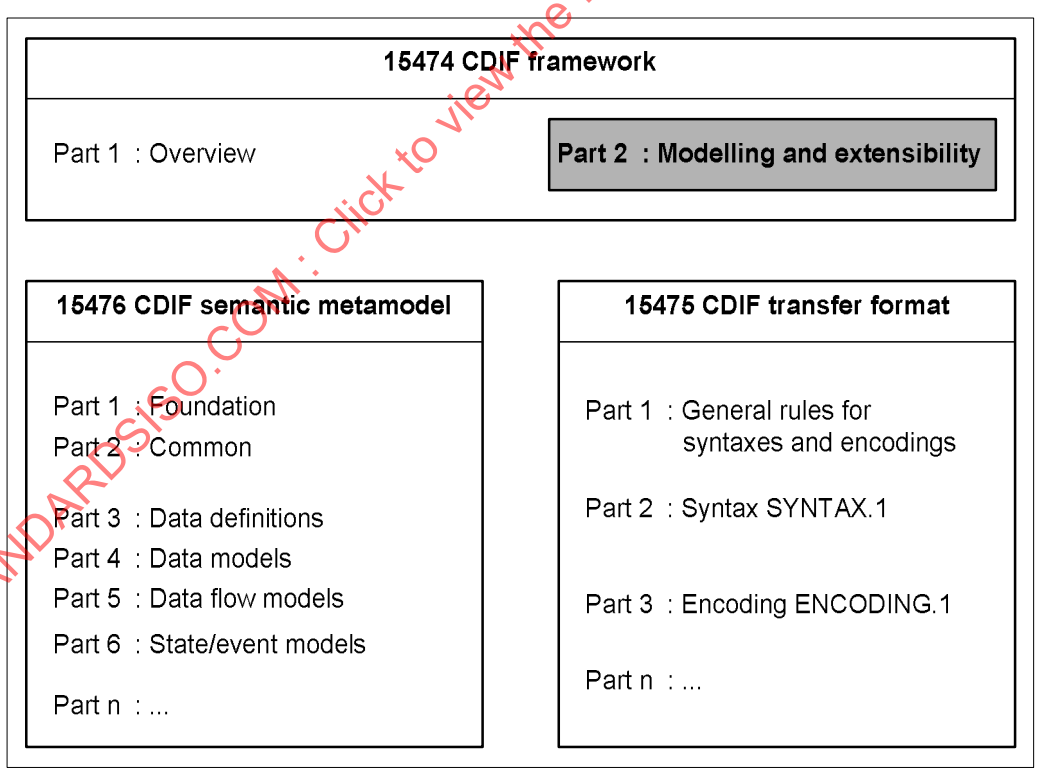

Figure 1 – CDIF family of standards

The diagram in Figure 1 depicts the various standards that comprise the CDIF family of standards. The shaded box depicts this Standard and its position in the CDIF family of standards.

This document (ISO/IEC 15474-2:2002, *Information technology - CDIF framework - Part 2: Modelling and extensibility*) includes the definition of the CDIF meta-metamodel and describes the rules and notations used throughout the CDIF family of standards. The rules for extending the CDIF semantic metamodel are also defined.

This document is intended to be used by anyone wishing to understand and/or use CDIF. This document provides an introduction to the entire CDIF family of standards. It is suitable for:

— Those evaluating CDIF,

— Those who wish to understand the principles and concepts of a CDIF transfer, and

— Those developing importers and exporters.

The document, ISO/IEC 15474-1:2002, *Information technology - CDIF framework - Part 1: Overview,* and this framework document ISO/IEC 15474-2:2002, *Information technology - CDIF framework - Part 2: Modelling and extensibility*, should be read first when initially exploring CDIF and before attempting to read other documents in the CDIF family of standards.

While there are no specific prerequisites for reading this document, it will be helpful for the reader to have familiarity with the following:

— Entity-Relationship-Attribute modelling;

— Modelling (CASE) tools;

— Information repositories;

— Data dictionaries;

— Multiple meta-layer modelling.

2 Conformance

A product is CDIF architecture conformant if and only if it can, as a property of that product, represent the product's metamodel instances, and/or the product's metamodel using the concepts defined in the ISO/IEC 15474-2:2002, *Information technology — CDIF framework — Part 2: Modelling and extensibility* ("Framework document"), and all the concepts defined in the standard ISO/IEC 15476-1:2002, *Information technology — CDIF semantic metamodel — Part 1: Foundation* ("Foundation document"), and obeys all the constraints and rules for metamodels and meta-data defined in the Framework document, and obeys all the rules and constraints defined in the Foundation document. Conformance to the graphical notation as defined in the Framework document is not required.

3 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 15474. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 15474 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 15474-1:2002, *Information technology — CDIF framework — Part 1: Overview*

ISO/IEC 13238-1:—[1)], *Information technology — Data management export/import — Part 1: Standardization framework*

ISO/IEC 15476-1:2002, *Information technology — CDIF semantic metamodel — Part 1: Foundation*

1) To be published.

4 Terms and definitions

For the purposes of this part of ISO/IEC 15474, the following definitions apply. Unless otherwise noted, the definitions are specific to this part of ISO/IEC 15474.

4.1 From other standards

4.1.1 ISO/IEC 15474-1

This part of ISO/IEC 15474 makes use of the following terms defined in ISO/IEC 15474-1:

Arity
Associative entity
Attributed relationship
Cardinality
CDIF
CDIF family of standards
CDIF graphical notation
CDIF semantic metamodel
CDIF meta-metamodel
Characteristic entity
Constraint
Encoding
Entity
Instance
Semantic metamodel
Kernel entity
Meta-attribute
Meta-entity
Meta-meta-attribute
Meta-meta-entity
Meta-metamodel
Meta-meta-relationship
Metamodel
Meta-object
Meta-relationship
Model
Model layers
N-ary relationship
Relationship
Role
Subject area
Syntax
Working metamodel

4.1.2 ISO/IEC 13238-1

This part of ISO/IEC 15474 makes use of the following terms from ISO/IEC 13238-1:

Transfer file
CDIF transfer file
Exporter
Importer

4.2 For this standard

For the purpose of this part of ISO/IEC 15474 new terms are defined when introduced. Double quotes are used to introduce new terms (e.g., "model layer").

5 Symbols (and abbreviated terms)

5.1 Naming and diagramming conventions

Conventions for naming, diagramming, describing and defining meta-objects can be found in Clause 7 of this document. Unless otherwise noted, they are applicable to meta-meta-objects named, diagrammed, described and defined in this document.

5.2 Abbreviations

The following abbreviations are used in this international standard:

CDIF CASE Data Interchange Format (originally)

ERA Entity-Relationship-Attribute modelling

6 CDIF meta-metamodel concepts and facilities

6.1 Introduction

This clause describes the concepts of the CDIF meta-metamodel. An overview and a diagram of the CDIF meta-metamodel are included in Clause 8 Meta-metamodel overview. A summary of the meta-metamodel is included in Clause 9 Meta-metamodel summary. The formal definition can be found in Clause 10 Meta-metamodel specification.

The CDIF meta-metamodel is defined using the same facilities that are used to define the CDIF metamodel (and could be used to define models). It is, therefore, quite common to get the names of the objects at the different meta-layers confused. To minimize this problem, the reader should remember the following:

- The components of the meta-metamodel are known as meta-meta-objects (i.e., meta-meta-entities, meta-meta-relationships and meta-meta-attributes);
- The components of the metamodel are known as meta-objects (i.e., meta-entities, meta-relationships and meta-attributes);
- The components of models are known as objects (e.g., processes, entities, flows);
- The names of the components in the meta-metamodel usually have a single meta- as a prefix (e.g., there is a meta-meta-entity, in the meta-metamodel, known as *MetaAttribute*);
- The names of the components in the metamodel usually have no meta prefix.

The CDIF meta-metamodel is the definition of the concepts used to develop CDIF-conformant metamodels. This includes the Entity-Relationship-Attribute diagramming technique and information definition requirements pertinent to the metamodel development process.

6.2 Modelling approach

6.2.1 Introduction

Because the CDIF meta-metamodel is used to define metamodels, it must be capable of expressing a complete description of a metamodel. A CDIF conformant metamodel contains three types of objects: entities, relationships and attributes; these objects in the metamodel are properly known as meta-entities, meta-relationships and meta-attributes. These objects are used to model the information content of the metamodel, and in fact, it is a rule in CDIF, that any single concept, or set of information may only be represented one way; that is, although there may be several ways of modelling a concept, only one will be included in the model.

The meta-metamodel is composed of two things:

- An Entity-Relationship-Attribute model that is used to describe the CDIF metamodel. This Entity-Relationship-Attribute model is also self-describing (i.e., is defined using itself).

- A graphical notation for representing occurrences (i.e., instances) of the meta-metamodel. This notation is used in CDIF in two ways: to depict the contents of subject areas in the CDIF metamodel, and to depict the CDIF meta-metamodel itself.

It is important for the reader to remember that the rules in this clause apply only to the metamodels defined using the CDIF meta-metamodel. In particular, these rules do not relate to Entity-Relationship-Attribute models that are transferred during a CDIF transfer (see ISO/IEC 15476-4:—, *Information technology — CDIF semantic metamodel — Part 4: Data models* for further information).

This clause describes some general modelling rules that are part of the definition of the CDIF meta-metamodel. They are described with a meta- prefix. The rules apply to the CDIF semantic metamodel and all extensions defined using the extensibility mechanism. The rules are also adhered to by the CDIF meta-metamodel[2)].

6.2.2 Subject areas

The CDIF semantic metamodel is a single integrated model that is a description of the set of concepts and notations used to define an information model. This model is huge; therefore, it is partitioned into manageable sections by building views of the underlying metamodel. These partitions, or "subject areas" as they are known, group together collectable meta-objects (meta-entities, meta-relationships and meta-attributes) that address a particular aspect within the Systems Development Life Cycle (e.g., Data Flow Modelling), or that provide similar functionality (e.g., Data Definition). Meta-objects may be shared by many subject areas. They are defined in one subject area, and referred to in other subject areas.

6.2.3 Many-to-many meta-relationships

The CDIF meta-metamodel supports many-to-many meta-relationships.

6.2.4 Attributed meta-relationships

The CDIF meta-metamodel supports attributed meta-relationships.

6.2.5 Subtyping of meta-entities

The CDIF meta-metamodel provides support for subtyping (sometimes known as specialization) and its converse, supertyping (sometimes known as generalization).

In particular, a meta-entity may be defined to be a "subtype" of another meta-entity. The subtype meta-entity inherits all of the meta-attributes and meta-relationships of its immediate and indirect supertype meta-entities.

Supertyping is used as a classification mechanism to group meta-objects with a substantial level of common semantics: an importer can use this classification as an aid to deciding how to represent any meta-object that is not directly representable within the importing tool.

Subtype meta-entities have unique *CDIFMetaIdentifiers*; that is, the *CDIFMetaIdentifier* is not inherited from the parent. The *Description*, *Constraints*, *Usage*, and *Aliases* that were defined for the parent meta-entity are inherited. These textual fields may also be expanded in the local definition of the subtype meta-entity. Subtype meta-entities inherit all the meta-attributes that have been defined for the parent meta-entity, and all those of its antecedents. The characteristics of the meta-attribute (for example, *Name*) are unchanged when inherited by a subtype meta-entity. Additional, local, meta-attributes may be defined specific to the subtype meta-entity.

---

2) The CDIF modelling rules may be used at any meta-level; they may be used to create models.

In the same way that meta-attributes are inherited, any meta-relationships that are defined with the parent meta-entity as the source or destination are inherited by subtype meta-entities. None of the characteristics of the meta-relationship are changed by its use by a subtype; that is, the *Name*, *Cardinality*, *CDIFMetaIdentifier*, *Aliases*, *Constraints*, *Description*, and *Usage*, are unchanged. Notice that even the full name of the meta-relationship is unchanged and will contain the name of the supertype meta-entity rather than that of the subtype.

A summary of the inheritance rules for subtype meta-entities is shown in Table 1.

Table 1 – Inheritance of meta-meta-attribute values by subtype meta-entities

META-META-ATTRIBUTE	Inherited by Subtype Meta-entity
ALIASES	Yes, may be extended
CDIFMETAIDENTIFIER	No
CONSTRAINTS	Yes, may be extended
DESCRIPTION	Yes, may be extended
ISABSTRACT	No
NAME	No
TYPE	No
USAGE	Yes, may be extended

6.2.6 Multiple inheritance in meta-entities

A meta-entity may have multiple supertypes, allowing for "multiple inheritance". In this case, the subtype inherits all of the meta-attributes and meta-relationships of all of its supertypes (and their supertypes, etc.).

There are several situations that occur when using multiple-inheritance that either seem to, or do, present inheritance conflicts.

When several inheritance threads of a meta-entity converge at a common supertype, the inheriting meta-entity does not inherit multiple copies of the meta-attributes and meta-relationships from that point to the top of the hierarchy; it only inherits one copy. Figure 2 shows meta-entity ***A*** having two subtype meta-entities (***B*** and ***C***), which are both supertype meta-entities of meta-entity ***D***. Both local and inherited meta-attributes for all meta-entities are shown.

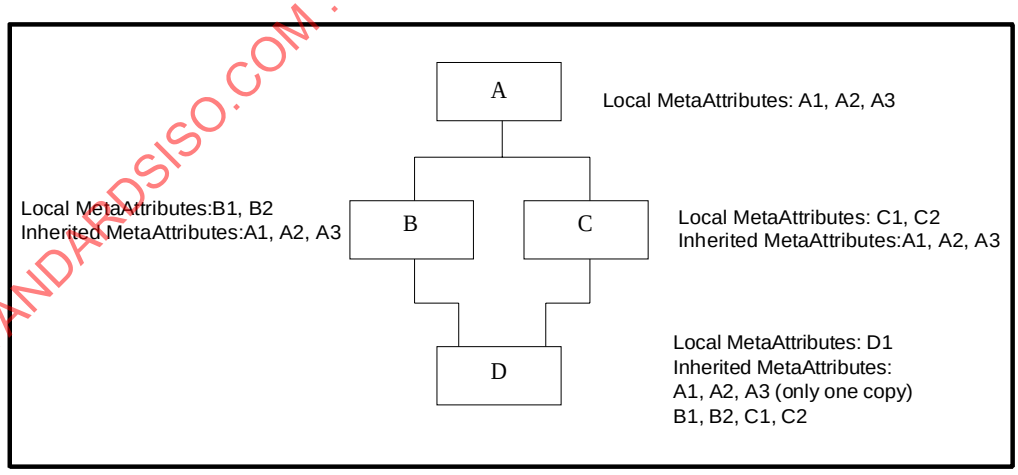

Figure 2 – Multiple inheritance

The other potential conflict occurs when a meta-entity inherits meta-attributes from two supertypes, both having a meta-attribute with the same name that are, in fact, different meta-attributes. The behaviour for CDIF is undefined in these circumstances. To guarantee proper (predictable) behaviour from importers and exporters, this situation

must be avoided. The CDIF semantic metamodel and CDIF meta-metamodel are defined ensuring that this does not occur.

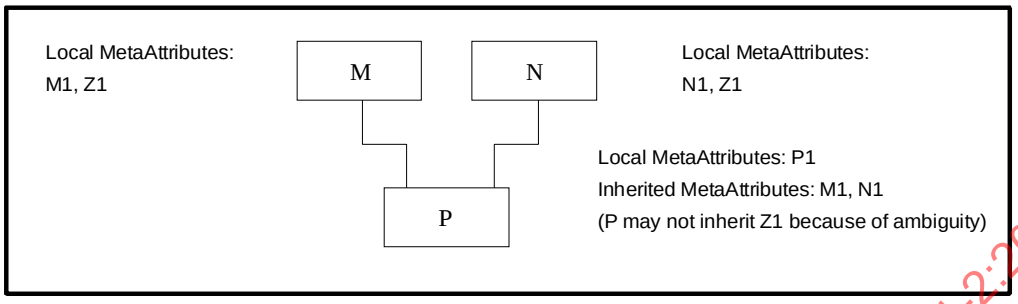

Figure 3 – Multiple inheritance: Invalid meta-attribute, *Z1*

Figure 3 shows an invalid inheritance example, where the meta-entity ***P*** has two supertype meta-entities (***M*** and ***N***) that both have an identically named meta-attribute, ***Z1***.

6.2.7 Subtyping of meta-relationships

A meta-relationship may be defined to be a subtype of another meta-relationship.

A subtype meta-relationship may be defined at the 'level' in the meta-entity hierarchy where the supertype meta-relationship was defined (that is, with the same source and destination meta-entities as the supertype). Alternatively, the source meta-entity or destination meta-entity, or both, may be subtypes of the source and destination meta-entities in the supertype meta-relationship.

Each subtype meta-relationship has a unique *CDIFMetaIdentifier*; that is, it does not inherit the *CDIFMetaIdentifier* from the supertype meta-relationship.

The cardinalities of the subtype meta-relationship are defined locally; however, they may only be as constrained, or more constrained than at the supertype level. Therefore, the minimum cardinality of the subtype must be greater than, or equal to, that of the supertype, and the maximum cardinality must be less than, or equal to, that of the supertype.

The other, textual, characteristics of the subtype meta-relationship are inherited from the supertype meta-relationship but may be further refined for the subtype.

The immediate and indirect subtype meta-relationship inherits any meta-attributes that have been defined for the supertype meta-relationship. None of the characteristics of the meta-attributes are changed. Additional meta-attributes may be created for the subtype meta-relationship.

A summary of the inheritance rules for subtype meta-relationships is shown in Table 2.

Table 2 – Inheritance of meta-meta-attribute values by subtype meta-relationships

META-META-ATTRIBUTE	Inherited by Subtype Meta-relationship
ALIASES	Yes, may be extended
CDIFMETAIDENTIFIER	No
CONSTRAINTS	Yes, may be extended
DESCRIPTION	Yes, may be extended
ISABSTRACT	No
MAXDESTCARD	No, must be locally defined
MAXSOURCECARD	No, must be locally defined
MINDESTCARD	No, must be locally defined
MINSOURCECARD	No, must be locally defined
NAME	No
USAGE	Yes, may be extended

Between any pair of meta-entities, there may be both a supertype meta-relationship (possibly due to inheritance by a subtype meta-entity) and a subtype meta-relationship defined. There may not be an instance of both the supertype meta-relationship and the subtype meta-relationship between the same pair of meta-entity instances. However, in most cases it is expected that the supertype meta-relationship will be abstract (not instantiable), which means that the condition of an instance of both supertype and subtype meta-relationships between the same instance pair will not arise.

6.2.8 Multiple inheritance of meta-relationships

Meta-relationships may inherit from multiple supertypes in the same way that meta-entities may inherit from multiple supertypes. The discussion in Clause 6.2.6, Multiple inheritance in meta-entities, that describes potential inheritance conflicts arising from multiple inheritance applies equally to multiple inheritance of meta-relationships, and the diagrams, although depicting meta-entities (in boxes) may be read as though the boxes represent meta-relationships.

6.2.9 Arity of meta-relationships[3)]

The CDIF semantic metamodel, and any extensions, must be defined using binary meta-relationships rather than n-ary meta-relationships. (Note that the functionality of n-ary relationships is covered in the Data Models subject area in the CDIF semantic metamodel, see ISO/IEC 15476-4:—, *Information technology — CDIF semantic metamodel — Part 4: Data models*.)

6.2.10 Mutual exclusivity of meta-relationships

In metamodels mutual exclusivity of meta-relationships is documented as a constraint on the meta-relationships that are in the exclusivity group.

6.3 Subject areas

6.3.1 Explicit and implicit use of *CollectableMetaObjects* in subject areas

Subject areas may contain some *CollectableMetaObjects* that are explicitly defined and/or used and, may contain some that are implicitly used through inheritance.

A meta-object is used in the subject area (and thus included in Clause 10, Specification of the subject area) when:

---

3) The arity of a relationship may also be called the degree of the relationship. A binary relationship has an arity of two; a ternary relationship has an arity of three, etc.

- It is defined in this subject area, or
- It has a local meta-attribute defined in this subject area, or
- It has a local subtype in this subject area (these are both supertype meta-entities whose meta-attributes and meta-relationships are inherited by meta-entities that are used, and supertype meta-relationships for those meta-relationships that are used), or
- It has a local meta-relationship in this subject area.

In the standardized CDIF subject areas, which are published as standards, the explicitly used *CollectableMetaObjects* are defined or referenced in Clause 10, Subject area specification, of each subject area standard. *CollectableMetaObjects* defined in the subject area are defined in Clause 10. *CollectableMetaObjects* used, but not defined in this subject area are referenced in clause 10.

When a new subject area is being created using extensibility, instances of the *CollectableMetaObject.IsDefinedIn.SubjectArea* and *CollectableMetaObject.IsUsedIn.SubjectArea* meta-meta-relationships must be created for all explicitly used *CollectableMetaObjects*. Instances of meta-meta-relationships should not be created for implicitly used *CollectableMetaObjects* since inheritance of meta-attributes and meta-relationships is assumed all the way up the inheritance hierarchies to *RootObject*.

6.3.2 Rules for defining subject area

Subject area views are built by instantiating the *CollectableMetaObject.IsDefinedIn.SubjectArea and CollectableMetaObject.IsUsedIn.SubjectArea* meta-meta-relationships for any *MetaEntity*, *MetaRelationship*, and *MetaAttribute* that are explicitly used (defined or referred to). The following rules should be observed when defining the scope of a subject area:

- Where a meta-relationship is explicitly used in a subject area, both the source and destination meta-entities must also be used.
- Where both source and destination meta-entities for a meta-relationship happen to be explicitly used in a subject area, there is no obligation to use the meta-relationship.
- Where a meta-entity is explicitly used in a subject area and it participates in another semantic metamodel in a mandatory meta-relationship (at the mandatory end), the meta-relationship need not be used in the subject area.
- Where a meta-attribute is explicitly used in a subject area, the meta-entity or meta-relationship that the meta-attribute describes must also be explicitly used in the subject area.
- Where a meta-entity or meta-relationship is explicitly used in a subject area, any local meta-attributes that describe it that are required by the subject area must be explicitly used.
- Where a meta-entity or meta-relationship is explicitly used in a subject area, there is no obligation to use any of its meta-attributes, even mandatory ones.
- All inherited meta-attributes for both meta-entities and meta-relationships will be implicitly used in the subject area all the way up the inheritance hierarchy. The *CollectableMetaObject.IsDefinedIn.SubjectArea* and *CollectableMetaObject.IsUsedIn.SubjectArea* meta-meta-relationship should not be instantiated for inherited meta-attributes.
- For subtype meta-entities, any meta-relationships in which the supertype meta-entity participates will be implicitly used in a subject area, provided the other meta-entity, or a subtype of it, is explicitly used.

  Note that where two subject areas are used in a transfer, a meta-relationship between two implicitly used supertype meta-entities (one in each subject area) will be inherited between the appropriate subtype meta-entities in the working metamodel for the transfer.

6.3.3 Working metamodel

A CDIF transfer may contain several, possibly overlapping subject areas. The actual subject areas, both standard and those created using extensibility, are identified in the transfer. An importer can then create what is called the working metamodel for a transfer by integrating the objects from standard subject areas and any additional objects created using extensibility.

When a transfer uses a standard subject area, the reference to that subject area is sufficient to define both explicitly and implicitly used meta-objects. This means that it is not necessary to give a subject area reference for the subject area where supertype meta-objects are defined purely to support those supertypes. If other meta-objects from that subject area are used in their own right, then the subject area reference must also be defined. In a CDIF transfer where the metamodel is being defined entirely using extensibility, a subject area reference for the Foundation subject area as defined in ISO/IEC 15476-1:2002*, Information technology - CDIF semantic metamodel - Part 1: Foundation*, must be used since it contains the definitions for the eventual roots of all meta-entity and meta-relationship hierarchies used in CDIF.

6.3.4 Subject area diagrams

Diagrams of subject areas will show explicitly used meta-entities. Meta-entities that are included in a diagram for clarification shall be shaded. Explicit meta-relationships will be shown in diagrams. Implicit meta-relationships (i.e. inherited meta-relationships) may be shown on the diagram even if the supertype meta-entity is not used explicitly. (See Clause 7.2 Meta-object graphical notation for further details.)

6.4 Metamodel extensibility

6.4.1 Introduction

Extensibility is an important aspect of the CDIF Family of Standards. In general, extensibility means allowing two (or more) tools that support CDIF to exchange information that is not defined in the CDIF semantic metamodel. Specifically, this involves having the exporter warn the importer about extensions to the standard CDIF semantic metamodel before transferring information that uses those extensions, see Clause 6.5, Exporter responsibilities. CDIF makes no statement about whether or not the importer needs to be able to retain the contents of information transferred (especially where extensibility has been used). For a more detailed discussion of the responsibilities of an importer, see Clause 6.6, Importer responsibilities. The following rules must be adhered to when using extensibility:

- Standardized subject areas may not be extended. This rule ensures that a consistent basis for importers to interpret is retained;
- A new, additional, subject area must be created to contain the extended meta-objects;
- The valid values of enumerated types are invariant by subject area. When an additional enumerated value is added it is added across the working metamodel, not for a specific subject area;
- Standardized meta-objects may also be used in exporter-defined subject areas;
- When reusing an existing meta-object in extensibility, the transfer must contain a reference to a subject area that contains the full definition of the meta-object;
- When any meta-object is added (i.e., a meta-entity, meta-relationship, meta-attribute, or subject area) a value must be defined for the meta-meta-attribute called *CDIFMetaIdentifier*. This identifier may have up to 128 characters but shall start with an alphabetic character and must be unique. The *CDIFMetaIdentifiers* of meta-objects in the CDIF semantic metamodel begin with a numeric character. No further structure to the identifier shall be given.

6.4.2 Extensibility

6.4.2.1 Add *SubjectArea*

An exporter may define a new subject area for the metamodel, provided the *CDIFMetaIdentifier* given to the new subject area is not already used by another attributable meta-object that has been created through extensibility in the transfer. The subject area may use both existing collectable meta-objects (i.e., meta-entities, meta-relationships, and meta-attributes) and newly defined ones. (See Clause 6.3.2, Rules for defining subject area for further details.)

6.4.2.2 Add *CollectableMetaObject.IsDefinedIn.SubjectArea* and *CollectableMetaObject.IsUsedIn.SubjectArea*

An exporter must define the collectable meta-objects that are used (defined or referred to) explicitly in a subject area. Inherited meta-attributes and meta-relationships are implicitly used and therefore this meta-meta-relationship shall not be defined for them. (See 6.3.2, Rules for defining subject area for further details.)

6.4.2.3 Add Meta-entities

An exporter may add a meta-entity, provided the *CDIFMetaIdentifier* given to the new meta-entity is not already used by another attributable meta-object that has been created through extensibility. All meta-entities must have *RootObject* (as defined in ISO/IEC 15476-1:2002*, Information technology — CDIF semantic metamodel — Part 1: Foundation*) as their eventual supertype, and therefore inherit the meta-attribute called *CDIFIdentifier*.

6.4.2.4 Add Meta-relationships

An exporter may add a meta-relationship, provided the *CDIFMetaIdentifier* given to the new meta-relationship is not already used by another attributable meta-object that has been created through extensibility. All meta-relationships must have *RootObject* (as defined in ISO/IEC 15476-1:2002*, Information technology — CDIF semantic metamodel — Part 1: Foundation*) as their eventual supertype and therefore inherit the meta-attribute called *CDIFIdentifier*. When adding a meta-relationship, the rules defined in Clause 3.4.4, Rules for Defining subject Area Scope, regarding the inclusion of meta-entities in subject areas must be observed.

6.4.2.5 Add *AttributableMetaObject.HasSubtype.AttributableMetaObject*

An exporter may define additional supertype associations for existing meta-entities and meta-relationships. Both subtype and supertype *AttributableMetaObjects* must be defined before the association may be specified. These *AttributableMetaObjects* may either be existing *AttributableMetaObjects* from the CDIF semantic metamodel, or may be created using extensibility. The supertype association is defined by an instance of the *AttributableMetaObject.HasSubtype.AttributableMetaObject* meta-meta-relationship. A new supertype association shall not be added to *RootObject*.

6.4.2.6 Add Meta-attribute to a Meta-entity or Meta-relationship

An exporter may add additional meta-attributes to a meta-entity or a meta-relationship.

6.4.2.7 Add Legal Enumerated Value to a Meta-attribute

An exporter may define additional legal values for a meta-attribute whose data type is enumerated. Enumerated values are not defined within the context of a subject area. New values, along with all the existing values are available to all subject areas in the working metamodel.

6.5 Exporter responsibilities

6.5.1 Introduction

This clause describes the responsibilities of a CDIF exporter. Note that this clause does not describe any syntax or encoding-specific responsibilities; these may be found in the definition of the syntax and encoding itself.

6.5.2 Extensibility

When an exporter extends the CDIF semantic metamodel, the extensions should not overlap semantically with the definitions contained in the semantic metamodel. The exporter shall minimize the use of extensibility by using standard definitions from the semantic metamodel where they exist. This ensures meaningful transfer between a set of importers and exporters.

Wherever extensibility is used the tool user cannot be assured that transfer of information between the exporter and the importer will provide meaningful results, unless there is mutual agreement on the extensions between one importer and one exporter.

6.5.3 Maximum output

All CDIF exporters should export as much information about a model as is possible. The exporter should not try to anticipate the capabilities of the importer (other than to include, or exclude, informed extensions to the standardized metamodel, that may be understood by some importers). This strategy ensures that the maximum amount of information will be transferred between the exporter and the importer.

6.5.4 Meta-attributes

For the instances of any meta-entities transferred by an exporter, values for all the mandatory (*IsOptional*: False) meta-attributes used in the subject area must be provided, even where no data is readily available.

6.6 Importer responsibilities

6.6.1 Introduction

This clause describes the responsibilities of a CDIF importer. Note that this clause does not describe any syntax or encoding-specific responsibilities; these may be found in the definition of the syntax and encoding itself.

6.6.2 Information retention

Any importer must be able to read all information contained in a CDIF transfer even if it doesn't do anything with the information once it has been read. This means that an importer must, for example, always parse a CDIF transfer correctly - even extensions. However, this does not mean the importer needs to do anything specific (e.g., retain) with the information once it has been accepted. The importer need not understand all the semantics defined within the CDIF semantic metamodel.

6.6.3 Working metamodel

While a CDIF transfer is being imported, a CDIF importer builds a "working metamodel", see Clause 6.3.3, Working metamodel. The initial working metamodel contains a single meta-object called *RootObject*. The working metamodel is then populated by the importer as it encounters and processes certain types of information in the transfer. Examples of transfer contents that populate the working metamodel include references to versions of the predefined CDIF semantic metamodel subject areas, the addition of a new *MetaEntity*, and so on.

When a CDIF importer encounters a reference to a meta-object in the metamodel, that meta-object must already exist in the current working metamodel. Forward references in the metamodel Clause of a CDIF transfer are not permitted.

Note that the working metamodel may be only a conceptual device and not implemented explicitly in an importer. In some cases, where the semantic metamodel has been used as the internal schema by the tool, mapping the components of a CDIF transfer directly to a tool's internal schema will be possible.

6.6.4 Meta-attributes

There are no default values specified in the CDIF semantic metamodel. Therefore, when an importer encounters an optional meta-attribute for which no value is given, it should interpret this lack of data as meaning that there was no data available to the exporter.

7 Modelling conventions

7.1 Naming of meta-objects

7.1.1 Names

All meta-objects and meta-meta-objects in CDIF (in metamodels and meta-metamodels) are named by concatenating all the words that name the meta-object or meta-meta-object; the first letter of each word is upper-case, the rest are lower-case (e.g., *MetaAttribute*, *AttributeDerivation*, *IsDrawnUsing*, *IsOptional*). This convention can be relaxed in those cases where the "words" that the object name is composed of, already have capitalization or word-separator information within them. This is especially common when one of the "words" is an acronym (e.g., *VCRModelName*).

Names shall conform to the Identifier data type, see Clause 10.1.6, Identifier.

In addition, the following conventions apply:

- It shall consist of the characters A-Z, a-z, 0-9, underscore (_) and hyphen (-);
- It shall start with an alphabetic character (i.e., A-Z or a-z);
- It shall end with an alphabetic character (i.e., A-Z or a-z) or a numeric character (i.e., 0-9).

7.1.2 Uniqueness of names

Names are not necessarily unique across all *MetaObjects*. Uniqueness of names within the working metamodel is considered for *SubjectAreas*, *MetaEntities*, *MetaRelationships* and *MetaAttributes* separately:

- The name given to each subject area must be unique within a working metamodel.
- The name given to each meta-entity must be unique within a working metamodel.
- The full name of each meta-relationship must be unique within a working metamodel, where the full name is the concatenation of the source meta-entity name, the meta-relationship name, and the destination meta-entity name.
- The name of a meta-attribute concatenated with the name of the *AttributableMetaObject* it defines must be unique within a working metamodel.

7.2 Meta-object graphical notation

7.2.1 Introduction

The same graphical notation is used for both the CDIF meta-metamodel and CDIF metamodel. The notation is described in this clause with regard to metamodels, but applies equally to the meta-metamodel defined in this

document[4]. For example, the meta-entity symbol is defined; however, the definition is equally valid for the meta-meta-entity symbol.

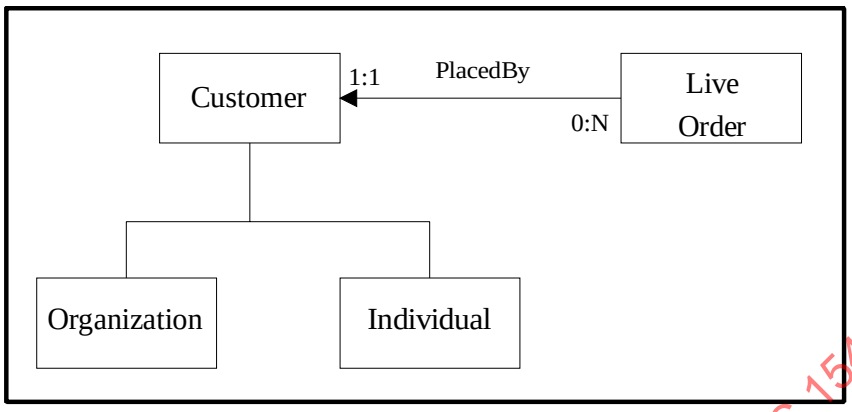

Figure 4 – Example of CDIF graphical notation

Figure 4 is an example model that uses the CDIF graphical notation.

7.2.2 Meta-entity symbol

Meta-entities (that is, instances of the meta-meta-entity called *MetaEntity*) used in a subject area are drawn as rectangular boxes, with the name of the meta-entity on a white background. Examples are ***'Meta-entity A'*** in Figure 5, and ***Customer*** in Figure 4.

Additionally, a special notation is used to represent some meta-entities on diagrams of metamodels. If the meta-entity is used in a subject area but is not defined in the subject area and is included in the diagram for clarity, the box shall be shaded. An example is ***'Meta-entity B'*** in Figure 5.

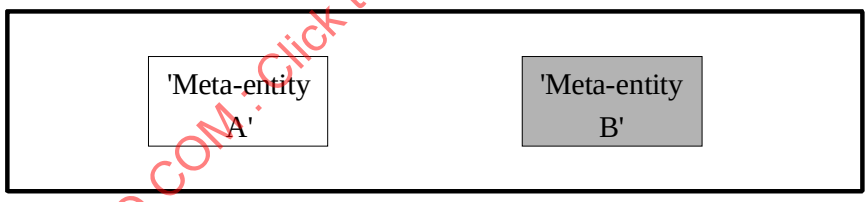

Figure 5 – The meta-entity symbol

7.2.3 Meta-relationship symbol

This symbol names and depicts an instance of the meta-meta-entity *MetaRelationship* that exists between two meta-entities. It is usually shown as a straight line (possibly multiple segments) joining two meta-entities/rectangles. A single arrowhead must appear somewhere along the meta-relationship line to indicate the direction in which the meta-relationship name shall be read. The two meta-entities participating in a meta-relationship are referred to as the "source" and "destination" as indicated by the arrowhead. Examples of meta-relationships are ***'Meta-relationship A'*** in Figure 6, and ***PlacedBy*** in Figure 4.

[4] CDIF Graphical Notation can be used for any meta-level. This particular example, shown in Figure 4, is of a model.

Figure 6 – The meta-relationship symbol

Additionally, three special notations may be used to represent meta-relationships on diagrams of metamodels.

If a meta-relationship is included in the diagram for clarification, is used in the subject area, but not defined in the subject area, the text of the meta-relationship name shall be back-shaded. This is shown for ***'Meta-relationship B'*** in Figure 6.

If a supertype meta-entity that participates in a meta-relationship is not included in the diagram, the inherited meta-relationship may be shown on the diagram as a broken line to/from the inheriting subtype meta-entity. This is shown for ***'Meta-relationship C'*** in Figure 6.

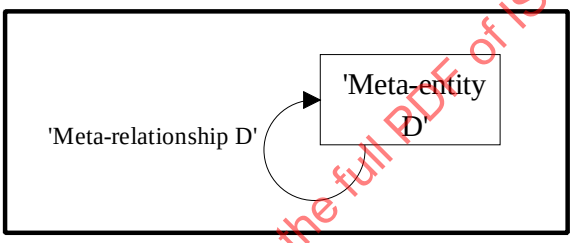

Figure 7 – A Reflexive meta-relationship

Where a reflexive (recursive) meta-relationship is drawn from a meta-entity to itself, it may be drawn as an arc (often referred to as a pig's ear or scorpion) as shown in Figure 7.

7.2.4 Meta-relationship cardinality symbol

The cardinality and optionality of each meta-relationship is defined by colon-separated labels (x:y) at both ends of the meta-relationship line symbol. The labels indicate the minimum (x) and the maximum (y) number of instances of the meta-entity nearest the label from the perspective of a single instance of the other meta-entity participating in the meta-relationship. The cardinalities of the ***PlacedBy*** meta-relationship are shown in Figure 8.

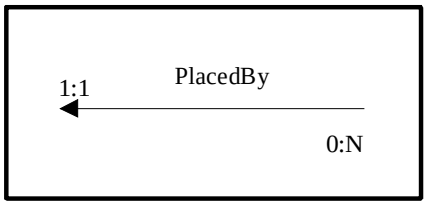

Figure 8 – The Meta-relationship cardinality symbol

The expansion of the notation into text for the ***PlacedBy*** meta-relationship shown in Figure 8 can be read:

"Each ***LiveOrder*** must be ***placed by*** one (and only one) ***Customer***."

"must be" because the minimum cardinality at the ***Customer*** end of the meta-relationship is 1, and "one (and only one)" because the maximum cardinality is also 1.

Reading the meta-relationship in the other direction gives us:

"Each ***Customer*** may be ***placer of*** one or more ***Live Orders***."

"may be" because the minimum cardinality at the ***Customer*** end is zero, and "one or more" because the maximum cardinality at ***LiveOrder*** is N, for many.

7.2.5 Meta-entity subtype hierarchy symbol

A special type of symbol is used to show that a meta-entity is a subtype of another meta-entity. It is shown as a hierarchy from the supertype meta-entity to each subtype. The supertype will always occur closer to the top of the diagram than its subtypes. The lines that connect the supertype to its subtypes are **not** labelled and are constrained to vertical and horizontal. Figure 9 shows an example of a single supertype meta-entity with two subtype meta-entities; that is, customers may either be organizations or individuals.

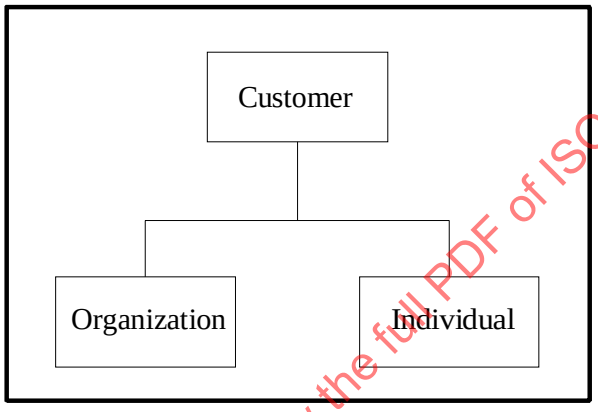

Figure 9 – The Meta-entity subtype hierarchy symbol

Figure 10 shows an example where a single subtype meta-entity is inheriting characteristics from two supertype meta-entities; an example of multiple inheritance.

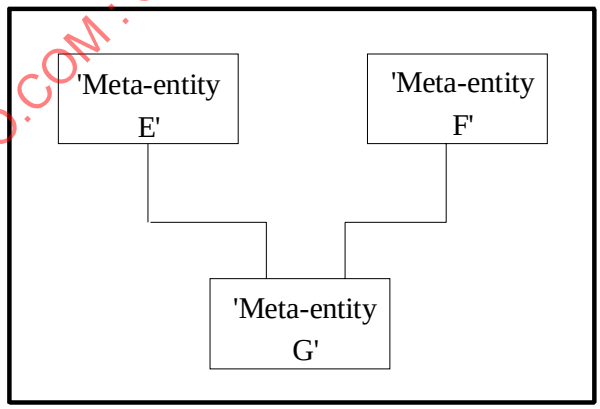

Figure 10 – The Meta-entity subtype hierarchy symbol: Multiple inheritance

7.2.6 Meta-relationship subtype hierarchy

The meta-relationship subtype hierarchy is not represented diagrammatically. The inheritance of meta-relationships is shown in the *AttributableMetaObject* hierarchy that is provided for each subject area. For example the following

meta-relationship hierarchy is defined in the Common subject area (ISO/IEC 15476-2:2002, *Information technology - CDIF semantic metamodel - Part 2: Common)*:

RootEntity.IsRelatedTo.RootEntity
    ProjectionComponent.IsProjectionOf.SemanticObject
        ProjectionComponent.IsFullProjectionOf.DataObject
        ProjectionComponent.IsProjectionOf.Attribute

7.2.7 Mutual exclusivity of meta-relationships

Mutual exclusivity of meta-relationships may be shown diagrammatically by placing an arc across the mutually exclusive meta-relationships.

Each instance of the meta-entity on which the mutual exclusivity group is specified can be connected to instances of only one of the meta-entities at the far end of all the included meta-relationships. If a connection to an instance of a meta-entity at the far end is required for each meta-entity instance at the near end, then the CDIF convention will be to indicate this with a minimum cardinality value of one (1:x) at the far end of all the included meta-relationships. This does not mean an instance of each is required; only one instance is required across the entire mutual exclusivity group. More than one instance can participate for the same meta-entity if the maximum cardinality is greater then one on the connecting meta-relationship.

Figure 11 shows that each ***ContactName*** must be for either an ***Organization*** or an ***Individual***, but that a single instance of ***ContactName*** would not be for both an ***Organization*** and an ***Individual***. (The "must be" can be deduced because of the minimum cardinality of one at the destination end of the ***For*** meta-relationship.)

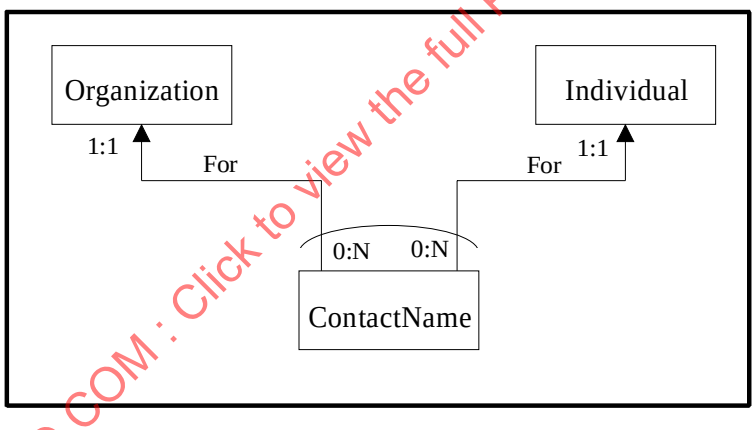

Figure 11 – Mutual exclusivity of meta-relationships

7.2.8 Instance diagrams

To aid in the understanding of some of the concepts modelled in this subject area, instance diagrams can be used. These diagrams differ from those described above in that an instance diagram depicts one or more occurrences or instances of meta-entities and meta-relationships depicted in the diagrams previously described. Where one of the diagrams previously described may show that the meta-model includes an *Automobile* meta-entity, an instance diagram might depict ***Honda***, ***Ford***, ***Mercedes***, and ***Fiat*** as instances of *Automobile*.

In an instance diagram, instances of meta-entities are depicted as boxes with a horizontal divider. Above the divider is the name of the meta-entity that is instantiated. Below the divider may be a name or a value assigned to a significant meta-attribute of that meta-entity. Where the name is inherited, inferred, or otherwise not significant to the example (e.g., some associative meta-entity instances), no name or value may be present.

Meta-relationships are depicted as arrows, just as in the previous diagrams described. They do not, however, have cardinality or exclusivity indicators, since none are necessary.

Meta-attributes of both meta-entities and meta-relationships are represented as:

- [meta-attribute name :] ***value***

where the name of the meta-attribute may be omitted depending on the ease of understanding the example with or without the name (e.g., *Name* might be omitted, *MinLength* might not).

When an instance of a meta-entity or meta-relationship not in this Subject Area is included for clarity, the box/arrow shall be shaded just as explained previously.

Figure 12 shows both the conceptual diagram and an instance diagram.

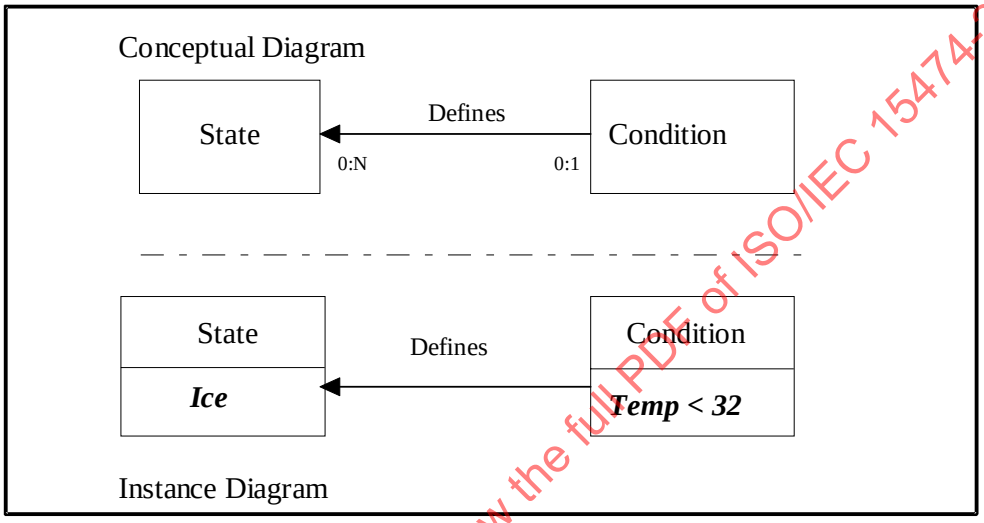

Figure 12 – Conceptual Diagram and an Instance Diagram

The names of objects and values for meta-attributes that appear in examples and instance diagrams are shown in bold italic (e.g., ***Customer, TRUE, 100***).

7.3 Meta-object definition format

7.3.1 Introduction

The same documentation is used for both the CDIF meta-metamodel and CDIF metamodel. The definition documentation is described in this clause with regard to metamodels, but applies equally (unless otherwise noted) to the meta-metamodel defined in this document. The definition Clause contains entries for all *MetaObjects* used in the subject area. Those defined in the subject area have definition entries. Those used, but not defined in the subject area have reference entries. This subclause defines the format for the definition entries.

7.3.2 Subject area definition

This is the formal definition of the instance of the meta-entity called *SubjectArea* (SA). It contains the information listed in Table 3.

Table 3 – Contents of subject area definition

SUBJECT AREA DEFINITION	
NAME	Subject area name (the name used to refer to the subject area). This is the name of the part of 15476 that defines the subject area, for instance "Data Models".
VERSIONNUMBER	The identifier assigned to the version of the subject area. This is the ISO/IEC part number of the document that defines the subject area, for instance "15476-4:2002".
CDIFMETAIDENTIFIER	A unique identifier for the subject area.
DESCRIPTION	The description of the subject area.
USAGE	Comments to clarify how the given subject area should be used by an importer/exporter writer.
ALIASES	Specifies alternate names for the given subject area by which the meta-object may be more normally known in any particular CASE environment.
CONSTRAINTS	Specifies restrictions on the meta-object, which can involve its interrelationships with one or more other objects.

7.3.3 Meta-entity definition

The definition of each meta-entity in the metamodel is structured according to the classes of information listed in Table 4.

For each meta-entity:

- Local subtypes, meta-relationships and meta-attributes are listed alphabetically.
- Detailed meta-attribute descriptions are listed alphabetically. These definitions are structured according to the classes of information listed in Table 5.

Table 4 – Contents of meta-entity definition

META-ENTITY DEFINITION	
NAME	The name used to refer to the meta-entity.
CDIFMETAIDENTIFIER	A unique identifier for the meta-entity (Not used in the meta-metamodel).
SUBJECTAREANAME	The name of the subject area where the meta-entity is defined (Not used in the meta-metamodel).
SUBJECTAREAVERSION	The version number of the subject area where the meta-entity is defined (Not used in the meta-metamodel).
DESCRIPTION	The description of the meta-entity.
USAGE	Comments to clarify how the given meta-entity should be used by an importer/exporter writer.
ALIASES	Specifies alternate names for the given meta-entity by which the meta-entity may be more normally known in any particular CASE environment.
CONSTRAINTS	Specifies restrictions on the meta-entity, which can involve its interrelationships with one or more other objects.
TYPE	A classification for the meta-entity that may convey additional semantic meaning that may be of use to the importer.
ISABSTRACT	If True, this indicate that this AttributableMetaObject will never directly be instantiated during a transfer.
LOCAL SUBTYPES	List of the meta-object names of the direct subtypes of the meta-entity.
LOCAL METARELATIONSHIPS	List of meta-relationships names defined for the meta-entity.
LOCAL METAATTRIBUTES	List of meta-attributes names defined for the meta-entity.

Full definitions of all local meta-attributes appear at this point. See Table 5 for format of local meta-attribute definitions.

7.3.4 Meta-attribute definition

The definition of each meta-attribute in the metamodel is structured according to the classes of information listed in Table 5.

Table 5 – Contents of meta-attribute definition

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF Name of the attributed meta-object
NAME	The name used to refer to the meta-attribute.
CDIFMETAIDENTIFIER	A unique identifier for the meta-attribute (Not used in the meta-metamodel).
SUBJECTAREANAME	The name of the subject area where the meta-attribute is defined (Not used in the meta-metamodel).
SUBJECTAREAVERSION	The version number of the subject area where the meta-attribute is defined (Not used in the meta-metamodel).
DESCRIPTION	The description of the meta-attribute.
USAGE	Comments to clarify how the given meta-attribute should be used by an importer/exporter writer.
ALIASES	Specifies alternate names for the given meta-attribute by which the meta-object may be more normally known in any particular CASE environment.
CONSTRAINTS	Specifies restrictions on the meta-attribute, which can involve its interrelationships with one or more other objects.
DATA TYPE	The type of data that can be stored in the meta-attribute.
DOMAIN	The pool of valid values for the meta-attribute.
LENGTH	The maximum length allowed for a meta-attribute of data type String.
ISOPTIONAL	Indicates whether or not a value must be supplied for the given meta-attribute for every instance of its *AttributableMetaObject.*

7.3.5 Meta-relationship definition

The definition for each meta-relationship in the metamodel is structured according to the classes of information listed in Table 6.[5)]

Table 6 – Contents of meta-relationship definition

META-RELATIONSHIP DEFINITION	
NAME	The name used to refer to the meta-relationship.
CDIFMETAIDENTIFIER	A unique identifier for the meta-entity (Not used in the meta-metamodel).
SUBJECTAREANAME	The name of the subject area where the meta-relationship is defined (Not used in the meta-metamodel).
SUBJECTAREAVERSION	The version number of the subject area where the meta-relationship is defined (Not used in the meta-metamodel).
DESCRIPTION	The description of the meta-relationship.
USAGE	Comments to clarify how the given meta-relationship should be used by an importer/exporter writer.
ALIASES	Specifies alternate names for the given meta-relationship by which the meta-relationship may be more normally known in any particular CASE environment.
CONSTRAINTS	Specifies restrictions on the meta-relationship, which can involve its interrelationships with one or more other objects.
ISABSTRACT	If True, this indicate that this *AttributableMetaObject* will never directly be instantiated during a transfer.
MINSOURCECARD	Minimum source meta-entity cardinality.
MAXSOURCECARD	Maximum source meta-entity cardinality.
MINDESTCARD	Minimum destination meta-entity cardinality.
MAXDESTCARD	Maximum destination meta-entity cardinality.
LOCAL SUBTYPES	List of the meta-relationship names of the direct subtypes of the meta-relationship.
LOCAL METAATTRIBUTES	List of meta-attributes names defined for the meta-relationship (Not used in the meta-metamodel).

Full definitions of all local meta-attributes appear at this point. See Table 5 for format of local meta-attribute definitions.
(Not used in the meta-metamodel)

[5)] Although the meta-metamodel permits inheritance in meta-meta-relationships, there are no instances in the meta-metamodel. Therefore, the definition of meta-meta-relationships does not use the line Local Subtypes.

7.4 Meta-object reference format

The specification Clause contains entries for all *MetaObjects* used in the subject area. Those defined in the subject area have definition entries. Those used, but not defined in the subject area have reference entries. This section specifies the format of reference entries.

Reference entries are not used in the meta-metamodel.

The definition of each meta-entity referenced in the metamodel is structured according to the classes of information listed in Table 7.

For each meta-entity:

- Local subtypes, meta-relationships and meta-attributes are listed alphabetically.
- Detailed meta-attribute descriptions are listed alphabetically. These definitions are structured according to the classes of information listed in Table 5.

Table 7 – Contents of meta-entity references

META-ENTITY REFERENCE	
NAME	The name used to refer to the meta-entity
CDIFMETAIDENTIFIER	A unique identifier for the meta-entity
SUBJECTAREANAME	The name of the subject area where the meta-entity is defined
SUBJECTAREAVERSION	The version number of the subject area where the meta-entity is defined
LOCAL SUBTYPES	List of the meta-object names of the direct subtypes of the meta-entity
LOCAL METARELATIONSHIPS.....	List of meta-relationships names defined for the meta-entity
LOCAL METAATTRIBUTES.....	List of meta-attributes names defined for the meta-entity

Full definitions of all local meta-attributes appear at this point. See Table 5 for format of local meta-attribute definitions.

The definition of each meta-relationship referenced in the metamodel is structured according to the classes of information listed in Table 8.

For each meta-relationship:

- Local subtypes, and meta-attributes are listed alphabetically.
- Detailed meta-attribute descriptions are listed alphabetically. These definitions are structured according to the classes of information listed in Table 5.

Table 8 – Contents of meta-relationship references

META-RELATIONSHIP REFERENCE	
NAME	The name used to refer to the meta-relationship
CDIFMETAIDENTIFIER	A unique identifier for the meta-relationship
SUBJECTAREANAME	The name of the subject area where the meta-relationship is defined
SUBJECTAREAVERSION	The version number of the subject area where the meta-relationship is defined
LOCAL SUBTYPES	List of the meta-object names of the direct subtypes of the meta-relationship
LOCAL METAATTRIBUTES.....	List of meta-attributes names defined for the meta-entity

Full definitions of all local meta-attributes appear at this point. See Table 5 for format of local meta-attribute definitions.

7.5 Meta-object summary format

7.5.1 Introduction

A summary Clause precedes the definition Clause. This Clause is informative, and introduces the subject area in a condensed format. It is made of the following components:

- *AttributableMetaObject* classification hierarchy;
- Meta-meta-entity summary;
- Meta-meta-relationship summary.

7.5.2 *AttributableMetaObject* classification hierarchy

The *AttributableMetaObject* classification hierarchy in the subject area summary sub-clause contains all the meta-entities and meta-relationships used in the subject area (and only those). Other supertypes of subtypes of objects shown in the hierarchy, where neither they nor any of their subtypes are used in this subject area, are not shown.

The *AttributableMetaObject* hierarchy is presented as an indented list of meta-object names. The indentation indicates the subtyping; all names indented below a given name are subtypes of that meta-object. For example, if ***Car*** and ***Truck*** are both subtypes of ***Vehicle***, this would be shown as follows:

Vehicle
    Car
    Truck

A meta-entity or meta-relationship and all its subtypes can be duplicated in the hierarchy due to multiple inheritance. Where this occurs, the duplicate entries are shown in *italics*.

The overall *AttributableMetaObject* supertype, *RootObject*, defined in ISO/IEC 15476-1:2002*, Information technology - CDIF semantic metamodel - Part 1: Foundation*, is shown at the top of the hierarchy.

Where a page break occurs in the list, all the supertypes of the first meta-entity or meta-relationship on the new page are repeated for clarity with a shaded background.

MetaObject name	SubjectArea name
Meta object name	Subject area
Subtype meta-object name	Subject area

7.5.3 MetaEntity and MetaRelationship summaries

Summaries are given of all the meta-entities and meta-relationships used in the subject area. The meta-entity definitions are given first, in alphabetical order, followed by the meta-relationship definitions, also in alphabetical order.

For each meta-entity used in the subject area, all the meta-attributes, both inherited and local, are listed. The inherited meta-attributes are shown first, in italics, sequenced alphabetically within parent meta-entity. The local meta-attributes are then listed alphabetically in normal font. The optionality, whether mandatory or optional, of each meta-attribute is given.

MetaEntity Name	
Inherited MetaAttribute Name	*Mandatory/Optional*
Local MetaAttribute Name	Mandatory/Optional

For each meta-relationship all the meta-attributes are listed sequenced alphabetically. Inherited and local meta-attributes are shown in the same way as for meta-entities. The optionality of each meta-attribute is given.

MetaRelationship Name	
Inherited MetaAttribute Name	*Mandatory/Optional*
Local MetaAttribute Name	Mandatory/Optional

8 Meta-metamodel overview

8.1 Introduction

This clause provides an overview to the CDIF meta-metamodel. The detailed definitions of the meta-meta-objects that form the meta-metamodel can be found in Clause 10, Meta-metamodel specification.

8.2 Meta-metamodel description

The CDIF meta-metamodel contains the definition and structure of data that may be included in CDIF-compliant metamodels.

MetaObject is the abstract name given to any thing that may appear in a metamodel, or in a meta-metamodel since it is defined in terms of itself. *MetaObject* has several meta-meta-attributes, including *CDIFMetaIdentifier* that must be unique for all *MetaObjects*. These meta-meta-attributes are inherited by all the meta-meta-entities in the meta-metamodel. *MetaObject* will never be directly instantiated; only as one of its direct or indirect subtypes. The two subtypes of *MetaObject* are *SubjectArea* and *CollectableMetaObject*.

SubjectAreas are groupings of *MetaObjects* that address either or both of the following:

- Some functional area or technique found within the Systems Development Life Cycle;
- Areas that provide similar functionality.

CollectableMetaObjects are those *MetaObjects* that may be grouped into a *SubjectArea*; they may either be *MetaAttributes*, or *AttributableMetaObjects*.

AttributableMetaObjects (that is *RootObject*, *MetaEntities* and *MetaRelationships*) may have *MetaAttributes*. *MetaAttributes* are the properties, or characteristics of *AttributableMetaObjects*. Both *MetaEntities* and *MetaRelationships* may have subtypes for inheritance.

8.3 Meta-metamodel diagram

Figure 13 shows a diagram of the CDIF meta-metamodel.

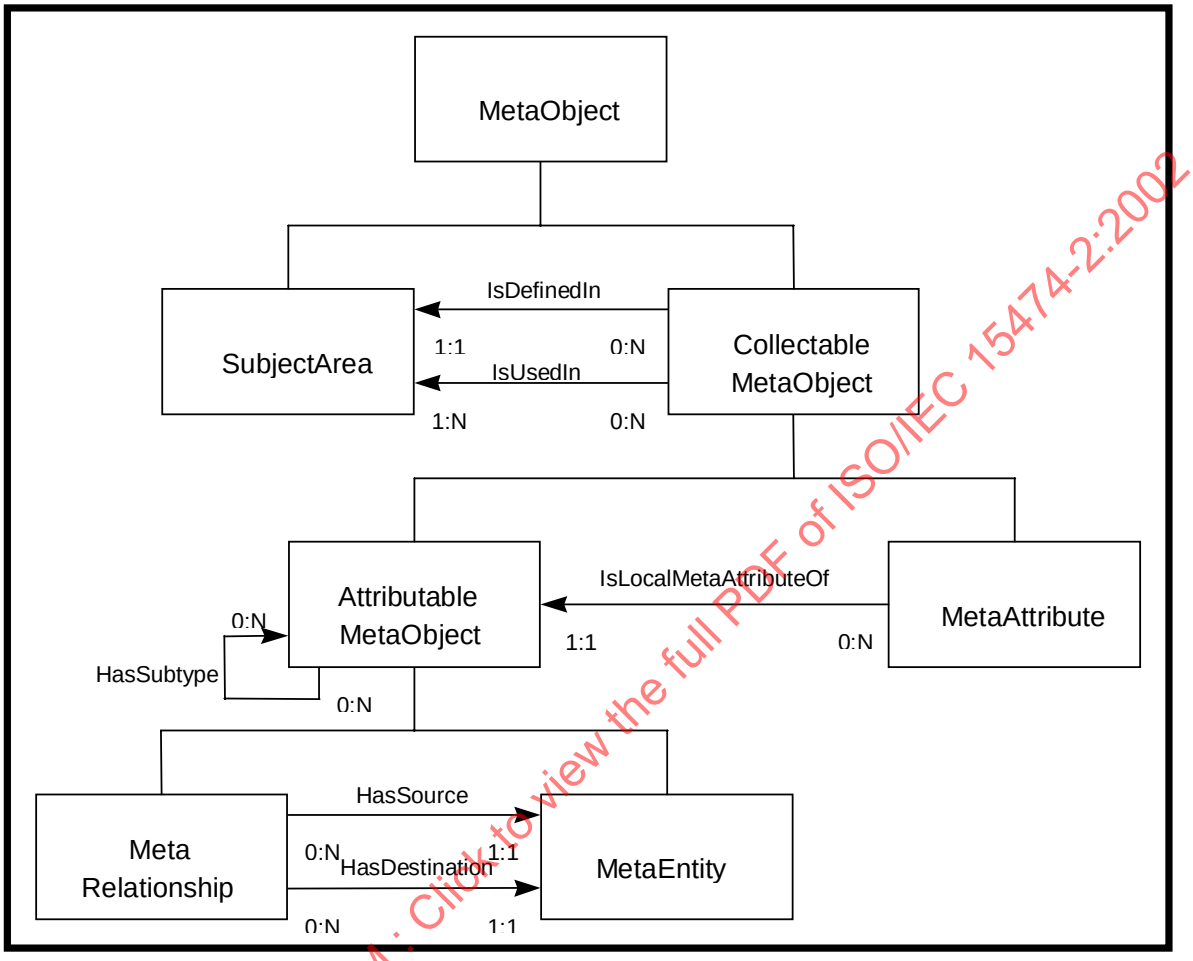

Figure 13 – The CDIF meta-metamodel

9 Meta-metamodel summary

9.1 *AttributableMetaObject* classification hierarchy [6)]

9.1.1 Introduction

The *AttributableMetaObject* hierarchy in this clause contains all the meta-meta-entities and meta-meta-relationships defined in the CDIF meta-metamodel.

The *AttributableMetaObject* hierarchy is presented as an indented list of meta-meta-entity and meta-meta-relationship names. The indentation indicates the subtyping; all names indented below a given name are subtypes

[6)] The term "*AttributableMetaObject*" rather than "Meta-meta-object" can be used here since, as has been said, the CDIF meta-metamodel is defined in terms of itself; that is, meta-meta-entities and meta-meta-relationships are *AttributableMetaObjects*.

of that meta-meta-object. For example, *AttributableMetaObject* and *MetaAttribute* are both subtypes of *CollectableMetaObject*, which is shown as follows:

CollectableMetaObject
    AttributableMetaObject
    MetaAttribute

9.1.2 Classification hierarchy

Meta-meta-object name	SubjectArea Name (not used)
MetaObject	
SubjectArea	
CollectableMetaObject	
AttributableMetaObject	
MetaEntity	
MetaRelationship	
MetaAttribute	
AttributableMetaObject.HasSubtype.AttributableMetaObject	
CollectableMetaObject.IsDefinedIn.SubjectArea	
CollectableMetaObject.IsUsedIn.SubjectArea	
MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject	
MetaRelationship.HasDestination.MetaEntity	
MetaRelationship.HasSource.MetaEntity	

9.2 Meta-meta-entity summary

AttributableMetaObject	
Aliases	*Optional*
CDIFMetaIdentifier	*Mandatory*
Constraints	*Optional*
Description	*Mandatory*
Name	*Mandatory*
Usage	*Optional*
IsAbstract	Optional

CollectableMetaObject	
Aliases	*Optional*
CDIFMetaIdentifier	*Mandatory*
Constraints	*Optional*
Description	*Mandatory*
Name	*Mandatory*
Usage	*Optional*

MetaAttribute	
Aliases	*Optional*
CDIFMetaIdentifier	*Mandatory*
Constraints	*Optional*
Description	*Mandatory*
Name	*Mandatory*
Usage	*Optional*
DataType	Mandatory
Domain	Optional
IsOptional	Mandatory
Length	Optional

MetaEntity	
Aliases	*Optional*
CDIFMetaIdentifier	*Mandatory*
Constraints	*Optional*
Description	*Mandatory*
IsAbstract	*Optional*
Name	*Mandatory*
Usage	*Optional*
Type	Optional

MetaObject	
Aliases	Optional
CDIFMetaIdentifier	Mandatory
Constraints	Optional
Description	Mandatory
Name	Mandatory
Usage	Optional

MetaRelationship	
Aliases	*Optional*
CDIFMetaIdentifier	*Mandatory*
Constraints	*Optional*
Description	*Mandatory*
IsAbstract	*Optional*
Name	*Mandatory*
Usage	*Optional*
MaxDestCard	Mandatory
MaxSourceCard	Mandatory
MinDestCard	Mandatory
MinSourceCard	Mandatory

SubjectArea	
Aliases	*Optional*
CDIFMetaIdentifier	*Mandatory*
Constraints	*Optional*
Description	*Mandatory*
Name	*Mandatory*
Usage	*Optional*
VersionNumber	Mandatory

9.3 Meta-meta-relationship summary

AttributableMetaObject.HasSubtype.AttributableMetaObject

CollectableMetaObject.IsDefinedIn.SubjectArea

CollectableMetaObject.IsUsedIn.SubjectArea

MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject

MetaRelationship.HasDestination.MetaEntity

MetaRelationship.HasSource.MetaEntity

10 Meta-metamodel specification

The meta-meta-object specifications in this clause are the formal definitions of the meta-meta-objects in the CDIF meta-metamodel. These take precedence over any descriptive or explanatory text in other parts of the document.

10.1 Data types specification

10.1.1 Introduction

This clause describes the data types that are available for the meta-attributes of meta-entities and meta-relationships, in all instances of the CDIF meta-metamodel (i.e., the CDIF metamodel, the CDIF meta-metamodel itself, and any additional metamodels built using the extensibility facilities of CDIF). The reader is reminded that the encoding of the information represented by these data types is defined in the specific syntax and encoding being used and is not defined here (e.g., a Boolean is defined here to hold either true or false, not ***1*** or ***0***, not ***Y*** or ***N***). The explicit limits for some of the data types are defined by the syntax being used.

Although some of the data types are defined as structured types, none of the components are optional. There are no defaults defined for any data types or for any of the components.

Table 9 shows the data types that are available for meta-meta-attributes in the meta-metamodel and for meta-attributes in metamodels.

Table 9 – Data types supported by CDIF

DATA TYPE	Description
BOOLEAN	TRUE or FALSE
DATE	A date (relative or absolute)
ENUMERATED	A list of valid values
FLOAT	A floating point number
IDENTIFIER	A name of an object
INTEGER	A signed integer
POINT	A location in three dimensions
STRING	A group of printable characters
TEXT	A group of characters, not necessarily printable
TIME	A time (relative or absolute)

Where a domain is not specified for an attribute, it shall be assumed to be the full range of values valid within the specified data type.

10.1.2 Boolean

The Boolean data type is used to represent a single item that can hold one of two values: TRUE or FALSE.

10.1.3 Date

The date data type is used to represent a date to the unit of granularity of a day. The class is specified to indicate whether the date is absolute or relative. An absolute date is given as ***Absolute***. Relative dates are either positive or negative to their (implied) point of reference; this is indicated with the ***RelativePositive*** and ***RelativeNegative*** classes.

10.1.4 Enumerated

An enumerated data type specifies a list of possible values for a meta-attribute. There is no implicit ordering or sequence. Each value must conform to the rules of a valid identifier data type.

10.1.5 Float

A decimal number with up to 16 decimal digits of precision within the range 10^{-1023} to 10^{1023}.

10.1.6 Identifier

The identifier data type is a string between one and 128 characters long. If the character set identified in the transfer envelope contains multibyte characters, they may be used in the names of model objects being transferred.

Identifiers are case-sensitive, that is "aaa" is different from "AAA".

When identifiers are used as the names of CDIF meta-objects, special rules apply. See Clause 7.1 Naming of meta-objects.

10.1.7 Integer

An integer is a signed number capable of holding values in the range from -2147483648 to 2147483647 (-2^{31} to $2^{31}-1$).

10.1.8 Point

A point is a triple of (x, y, z) coordinates and represents an address in the coordinate frame. Each coordinate is represented using an integer.

10.1.9 String

The string data type is represented as a group of printable characters of some maximum length, where 1 <= maximum length <= 1024.

10.1.10 Text

The text data type is represented as a group of characters, not necessarily printable, of any pre-determined maximum length.

10.1.11 Time

The time data type is used to represent a time to the granularity of fractional seconds with a class to indicate whether the time is absolute or relative. Absolute times are given either as ***AbsoluteUTC***[7] or ***AbsoluteLocal***. Relative times are either positive or negative relative to their (implied) point of reference; this is indicated with the ***RelativePositive*** and ***RelativeNegative*** classes.

Times recorded in a CASE tool are exported as the values recorded by the tool. No conversions are specified by CDIF for time values where more than one time zone is involved.

[7] UTC stands for Universal Coordinated Time (English translation of French term) as defined in recommendation 460 of the Consultative Committee on International Radio (CCIR) [1].

10.2 Meta-meta-entity and Meta-meta-attribute specifications

10.2.1 AttributableMetaObject

META-META-ENTITY DEFINITION	
NAME	**AttributableMetaObject**
CDIFMETAIDENTIFIER	N/A
SUBJECTAREANAME	N/A
SUBJECTAREAVERSION	N/A
DESCRIPTION	Defines an abstract supertype for those meta-meta-objects (i.e., *MetaEntities* and *MetaRelationships*) that may have local meta-attributes.
USAGE	Used to define meta-meta-attributes and meta-meta-relationships that are common to both *MetaEntity* and *MetaRelationship*.
ALIASES	
CONSTRAINTS	This meta-meta-entity has only one instance: *RootObject*.
TYPE	Kernel
ISABSTRACT	False
LOCAL SUBTYPES	MetaEntity MetaRelationship
LOCAL METARELATIONSHIPS	AttributableMetaObject.HasSubtype.AttributableMetaObject MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject
LOCAL METAATTRIBUTES	IsAbstract

META-META-ATTRIBUTE DEFINITION	META-META-ATTRIBUTE OF AttributableMetaObject
NAME	**IsAbstract**
CDIFMETAIDENTIFIER	N/A
SUBJECTAREANAME	N/A
SUBJECTAREAVERSION	N/A
DESCRIPTION	When True, this indicate that the *AttributableMetaObject* will never directly be instantiated.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Boolean
DOMAIN	
LENGTH	
ISOPTIONAL	False

10.2.2 CollectableMetaObject

META-META-ENTITY DEFINITION	
NAME	**CollectableMetaObject**
CDIFMETAIDENTIFIER	N/A
SUBJECTAREANAME	N/A
SUBJECTAREAVERSION	N/A
DESCRIPTION	Defines an abstract supertype for those meta-meta-objects (i.e., *AttributableMetaObjects* and *MetaAttributes*) that may be grouped into subject area views.
USAGE	Used to group those meta-objects that may be used in subject areas.
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	True
LOCAL SUBTYPES	AttributableMetaObject MetaAttribute
LOCAL METARELATIONSHIPS	CollectableMetaObject.IsDefinedIn.SubjectArea CollectableMetaObject.IsUsedIn.SubjectArea
LOCAL METAATTRIBUTES	

10.2.3 MetaAttribute

META-META-ENTITY DEFINITION	
NAME	**MetaAttribute**
CDIFMETAIDENTIFIER	N/A
SUBJECTAREANAME	N/A
SUBJECTAREAVERSION	N/A
DESCRIPTION	Defines a specific property of a meta-entity or meta-relationship (i.e., an *AttributableMetaObject*).
USAGE	
ALIASES	Property, Characteristic.
CONSTRAINTS	The names of all *MetaAttributes* must be unique within their parent object within the working metamodel. If the attribute is an attribute of a subtype, this uniqueness rule applies all the way up that particular inheritance thread. Within the CDIF semantic metamodel, all meta-attributes that are semantically identical are given the same names.
TYPE	Kernel
ISABSTRACT	False
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject
LOCAL METAATTRIBUTES	DataType Domain IsOptional Length

META-META-ATTRIBUTE DEFINITION	META-META-ATTRIBUTE OF MetaAttribute
NAME	**DataType**
CDIFMETAIDENTIFIER	N/A
SUBJECTAREANAME	N/A
SUBJECTAREAVERSION	N/A
DESCRIPTION	The type of data that can be stored in a meta-attribute.
USAGE	
ALIASES	Usage, Type
CONSTRAINTS	
DATA TYPE	Enumerated
DOMAIN	Identifier, Integer, String, Text, Boolean, Date, Time, Float, Enumerated, Point
LENGTH	
ISOPTIONAL	False

META-META-ATTRIBUTE DEFINITION	META-META-ATTRIBUTE OF MetaAttribute
NAME	**Domain**
CDIFMETAIDENTIFIER	N/A
SUBJECTAREANAME	N/A
SUBJECTAREAVERSION	N/A
DESCRIPTION	The pool of valid values for a meta-attribute.
USAGE	This is not a computable field. When the type of the meta-attribute is *Enumerated*, the domain shall contain a comma-separated list of unique, valid values for the meta-attribute. When the type of the meta-attribute is not *Enumerated*, the domain shall be expressed in unambiguous English. Where domain is not specified, the full range of values for the specified data type shall be assumed.
ALIASES	
CONSTRAINTS	The Domain must be specified where the data type is enumerated.
DATA TYPE	Text
DOMAIN	
LENGTH	
ISOPTIONAL	True

META-META-ATTRIBUTE DEFINITION	META-META-ATTRIBUTE OF MetaAttribute
NAME	**IsOptional**
CDIFMETAIDENTIFIER	N/A
SUBJECTAREANAME	N/A
SUBJECTAREAVERSION	N/A
DESCRIPTION	Indicates whether or not a value must be supplied for a given meta-attribute for every instance of its *AttributableMetaObject*.
USAGE	
ALIASES	NotRequired, NotMandatory, Nullable
CONSTRAINTS	
DATA TYPE	Boolean
DOMAIN	
LENGTH	
ISOPTIONAL	False

META-META-ATTRIBUTE DEFINITION	META-META-ATTRIBUTE OF MetaAttribute
NAME	**Length**
CDIFMETAIDENTIFIER	N/A
SUBJECTAREANAME	N/A
SUBJECTAREAVERSION	N/A
DESCRIPTION	The maximum length allowed for a meta-attribute of data type String.
USAGE	Used only for Strings.
ALIASES	Size
CONSTRAINTS	This should only be specified for a meta-attribute of data type String.
DATA TYPE	Integer
DOMAIN	1<=n=<1024
LENGTH	
ISOPTIONAL	True

10.2.4 MetaEntity

META-META-ENTITY DEFINITION	
NAME	**MetaEntity**
CDIFMETAIDENTIFIER	N/A
SUBJECTAREANAME	N/A
SUBJECTAREAVERSION	N/A
DESCRIPTION	*MetaEntity* defines the characteristics of all meta-entities that occur in a metamodel.
USAGE	
ALIASES	
CONSTRAINTS	The names of all *MetaEntities* must be unique within the meta-layer that contains them within the working metamodel. All *MetaEntities* must have *RootObject* as their eventual supertype. For subtype meta-entities, the *Description, Usage, Aliases*, and *Constraints* described local to this meta-entity are in addition to those of the supertype meta-entity. The meta-relationships in which the supertype meta-entity participated will be inherited by the subtype meta-entity. The meta-attributes of the supertype meta-entity are also inherited.
TYPE	Kernel
ISABSTRACT	False
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	MetaRelationship.HasDestination.MetaEntity MetaRelationship.HasSource.MetaEntity
LOCAL METAATTRIBUTES	Type

META-META-ATTRIBUTE DEFINITION	META-META-ATTRIBUTE OF MetaEntity
NAME	**Type**
CDIFMETAIDENTIFIER	N/A
SUBJECTAREANAME	N/A
SUBJECTAREAVERSION	N/A
DESCRIPTION	A classification for the meta-entity that may convey additional semantic meaning, which may be of use to the importer.
USAGE	Is used to indicate which modelling semantic is being represented by the meta-entity. Kernel: This classification is used for a meta-entity whose instances are meaningful without the occurrences of other meta-entities. Characteristic: Each instance of a characteristic meta-entity is logically only related to one instance of one other meta-entity. Therefore an importer could incorporate the meta-attributes of a characteristic meta-entity with those of the 'owning' meta-entity, where the owning meta-entity is the one to which the characteristic meta-entity is related with a cardinality of 1:1. Associative: An associative meta-entity is used to represent a meta-relationship between other meta-entities. It is used when a meta-relationship does not otherwise provide sufficient mechanisms.
ALIASES	Classification, represented semantic
CONSTRAINTS	
DATA TYPE	Enumerated
DOMAIN	Kernel, Characteristic, Associative
LENGTH	
ISOPTIONAL	True

10.2.5 MetaObject

META-META-ENTITY DEFINITION	
NAME	**MetaObject**
CDIFMETAIDENTIFIER	N/A
SUBJECTAREANAME	N/A
SUBJECTAREAVERSION	N/A
DESCRIPTION	Defines shared meta-meta-attributes of all meta-meta-entities. It is the root of the type hierarchy in the CDIF meta-metamodel.
USAGE	
ALIASES	
CONSTRAINTS	The *CDIFMetaIdentifier* is the "key" of the *MetaObject*, in the sense that each of its different values uniquely identifies each instance of the *MetaObject*.
TYPE	Kernel
ISABSTRACT	True
LOCAL SUBTYPES	CollectableMetaObject SubjectArea
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	Aliases CDIFMetaIdentifier Constraints Description Name Usage