enosema

Standard

EFS 2510:2026

Terminology management — Concept systems and concept relations

First edition

Reference number
EFS 2510:2026

© EFS 2026

(c) 2026 Enosema Foundation.

This document is made available under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. Unless otherwise specified, or required in the context of its implementation, you are free to use, share, and adapt this document, provided that appropriate credit is given to the Enosema Foundation and that any changes are indicated. The full license text is available at https://creativecommons.org/licenses/by/4.0/.

Enosema Foundation
2885 Sanford Ave Sw Suite 32361
Grandville MI 49418-1342
USA

E-mail: info@enosema.org

Published in the USA

© EFS 2026 – All rights reserved

Contents

Page

Foreword ..... **v**

Introduction ..... **vi**

1 Scope **1**

2 Normative references **1**

3 Terms, definitions and abbreviated terms **2**
3.1 Terms and definitions 2
3.2 Abbreviated terms 4

4 Concept systems **4**
4.1 General 4
4.2 Concepts and characteristics 4
4.3 Concept coordination 4
4.3.1 Generic coordination 4
4.3.2 Partitive coordination 5
4.3.3 Instantial coordination 5
4.4 Criteria of subdivision and multidimensionality 5
4.5 Concept systems as graphs 5

5 Concept relation vocabulary **6**
5.1 General 6
5.2 Hierarchical relations 6
5.3 Equivalence and mapping relations 7
5.4 Associative relations 7
5.5 Lexical, lifecycle, register and versioning relations 9
5.6 The relation vocabulary as a registered item 11

6 Binary relations **11**
6.1 When to use a binary relation 11
6.2 Recording direction and inverse 11
6.3 Assertions between designations 11
6.4 Informative versus normative relations 12

7 N-ary relations **12**
7.1 General 12
7.2 The binary-versus-n-ary decision rule 12
7.3 Partitive n-ary relations 13
7.4 Generic n-ary relations 13
7.5 Sequential n-ary relations 13
7.6 Criterion records 14
7.7 Registration of n-ary relations 14
7.8 Serialization of n-ary relations 14

8 Machine-readable representation **15**
8.1 General 15
8.2 Semantic content of a relation record 15
8.3 Mapping to SKOS 15
8.4 JSON-LD and typed representation 15
8.5 Relation identifiers in interchange 16

9 Quality constraints on concept systems **16**
9.1 General 16
9.2 Structural constraints 16
9.3 Constraint checking on lifecycle relations 16
9.4 Constraint checking on n-ary relations 17
9.5 Consistency of inverses 17
9.6 Review of concept systems 17

© EFS 2026 – All rights reserved

Bibliography **18**

© EFS 2026 – All rights reserved

Foreword

The Enosema Foundation ("ES") is the premier non-profit standardization organization for terminology standardization and terminology related processes such as management. It facilitates the education, standardization, research, promotion, definition, and usage of terminology resources and management practices globally.

ES works with international partners and experts across the globe, reflecting the international nature of its mission. More information about ES is available on the official website (https://www.enosema.org).

The procedures used to develop this document and those intended for its further maintenance are described in the ES Directives.

In particular, the different approval criteria needed for the different types of ES documents should be noted. This document was drafted in accordance with the editorial rules of the ES Directives.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ES 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 provided in the Introduction.

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

This document was prepared by Technical Committee **Terminology**.

© EFS 2026 – All rights reserved

Introduction

Concepts do not exist in isolation. A concept is understood through its relations to other concepts: the broader concept it is a kind of, the parts that fit together to constitute it, the instances that realize it, and the multitude of associative links that relate it to its causes, materials, actors, and products. The systematic arrangement of concepts according to their relations forms a **concept system**, and the quality of a concept system largely determines the quality of the terminology that is derived from it.

International standards establish the foundations of this work. ISO 704 establishes the principles and methods of terminology work, including the theory of concept systems and the characteristics of concepts. ISO 1087-1 provides the vocabulary of terminology work. ISO 25964, developed for thesauri and interoperability with other vocabularies, enumerates the semantic relationships that connect concepts. ISO 12620 provides a data category registry whose relation categories are widely implemented. ISO 10241-1 governs the drafting of terminological entries in standards. ISO 19135 establishes procedures for the registration of items, the foundation of the FERIN framework for the registration of information.

In practice, however, organizations that manage concepts face recurring questions that these standards answer only in fragments:

— which relation types should a concept system distinguish, and what is the semantics of each;

— when is a pairwise (binary) relation the right assertion, and when is a joint, multi-member assertion (a “rake”) required;

— how are completeness, per-member multiplicity, and the criterion of subdivision recorded so that concept diagrams and their meaning survive into machine-readable form;

— how are relations managed over time as concepts are extended, split, instantiated, or retired within a register.

This document answers these questions. It specifies the relation vocabulary of ES concept systems, the rules for choosing between binary relations and n-ary relations, the metadata that accompanies multi-member relations, and the quality constraints that keep a concept system coherent as it evolves. It is written to be directly implementable: each relation type is defined with its direction, its inverse, and its semantic conditions, so that terminology management systems can enumerate, validate, and interchange concept relations without ambiguity.

This document is part of a suite of ES specifications for terminology management. EFS 2501 addresses principles and methods of terminology management, EFS 2504 addresses multilingual terminology management, and EFS 2507 addresses the abstract model of concept relationships. This document builds on those foundations and specifies the relation vocabulary and system-level practices in full.

© EFS 2026 – All rights reserved

Terminology management — Concept systems and concept relations

1 Scope

This document specifies the concept systems and concept relations used in terminology management, for the purposes of the Enosema Foundation and of any organization that adopts this document.

It specifies:

— the structure of concept systems, including concept coordination (generic, partitive and instantial), criteria of subdivision, and multidimensionality;

— a controlled vocabulary of concept relations, enumerating each relation type with its semantics, direction, and inverse;

— the rules for choosing between binary relations and n-ary relations (hyperedges), and the metadata that n-ary relations carry, including completeness, per-member presence and count, delimiting status, and the criterion of sequencing;

— requirements for the machine-readable representation of concept relations and for their management within registers.

This document does not specify:

— the structure of terminological entries in standards, which is specified in ISO 10241-1 and applied in EFS 2501;

— the management of multilingual equivalence beyond its impact on relations, which is specified in EFS 2504;

— the abstract data model of concept relationships, which is specified in EFS 2507;

— the organizational requirements for operating a concept management system, which are specified in EFS 2513.

2 Normative references

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

ISO 704, *Terminology work — Principles and methods*.

ISO 1087-1[1)] , *Terminology work — Vocabulary — Part 1: Theory and application*.

ISO 25964-1, *Information and documentation — Thesauri and interoperability with other vocabularies — Part 1: Thesauri for information retrieval*.

ISO 25964-2, *Information and documentation — Thesauri and interoperability with other vocabularies — Part 2: Interoperability with other vocabularies*.

1) Withdrawn.

© EFS 2026 – All rights reserved

ISO 12620[2] , *Management of terminology resources — Data category specifications*.

ISO 10241-1, *Terminological entries in standards — Part 1: General requirements and examples of presentation*.

ISO 19135-1[3] , *Geographic information — Procedures for item registration — Part 1: Fundamentals*.

W3C SKOS, *SKOS Simple Knowledge Organization System Reference*, W3C Recommendation, https://www.w3.org/TR/skos-reference/

ISO/IEC Directives, Part 2, *ISO/IEC Directives, Part 2, Principles and rules for the structure and drafting of ISO and IEC documents*.

3 Terms, definitions and abbreviated terms

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

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

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

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

3.1 Terms and definitions

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

ISO and ES maintain terminological databases for use in standardization at the addresses given in the bibliography.

3.1.1
concept
unit of knowledge created by a unique combination of characteristics

Note 1 to entry: A concept is not necessarily bound to a particular language. It is, however, influenced by the social or cultural background which often leads to different categorizations.

3.1.2
concept system
set of concepts and the set of relations that hold among them

3.1.3
concept relation
semantic relation between two or more concepts within a concept system

[NOTE: In this document, a relation between exactly two concepts is a **binary relation** (3.1.4); a relation between a comprehensive concept and two or more coordinate concepts is an **n-ary relation** (3.1.5).]

3.1.4
binary relation
concept relation (3.1.3) that holds between exactly two concepts

3.1.5
n-ary relation
hyperedge

2) Withdrawn.

3) Withdrawn.

© EFS 2026 – All rights reserved

concept relation (3.1.3) that holds between a comprehensive concept and two or more coordinate concepts that are jointly significant with respect to the comprehensive concept

3.1.6
coordinate concept
concept to which a given *concept* (3.1.1) is related by a shared superordinate concept and a shared criterion of subdivision

[SOURCE: ISO 1087-1, modified — the shared criterion of subdivision is made explicit]

3.1.7
comprehensive concept
superordinate concept that subsumes two or more subordinate concepts which, taken together, characterize it

[NOTE: In a partitive relation, the comprehensive concept is the whole that the parts fitted together constitute; in a generic relation, it is the class that the kinds jointly constitute; in a sequential relation, it is the process or development that the ordered stages jointly constitute.]

3.1.8
criterion of subdivision
dimension
basis for the division of a concept system into subordinate concepts

3.1.9
multidimensionality
characteristic of a *concept system* (3.1.2) in which the concepts of one subdivision level are subdivided according to more than one *criterion of subdivision* (3.1.8)

3.1.10
completeness
indication of whether the members of an *n-ary relation* (3.1.5) exhaust the intension of the comprehensive concept

3.1.11
presence
indication of whether a member of an *n-ary relation* (3.1.5) exists in every instance of the comprehensive concept or only in some instances

3.1.12
delimiting characteristic
characteristic of a concept that distinguishes it from coordinate concepts within the same subdivision

[NOTE: In concept diagrams, the part of a partitive relation that delimits a subordinate concept is shown with a bold line of three-fold width.]

3.1.13
sequential relation
n-ary relation (3.1.5) connecting a comprehensive concept to two or more subordinate concepts ordered by a criterion of sequencing

3.1.14
register
set of information items with an assigned status maintained by a registration authority

3.1.15
registration authority
RA
organization that performs the administrative functions required to operate a *register* (3.1.14)

© EFS 2026 – All rights reserved

3.2 Abbreviated terms

ES	Enosema Foundation
RA	registration authority
SKOS	Simple Knowledge Organization System
TC	technical committee
URN	uniform resource name

4 Concept systems

4.1 General

Concepts are managed as systems, not in isolation. A concept system arranges concepts according to their relations so that each concept is positioned by its intension (its characteristics) and its extension (its instances), and each term assigned to a concept is motivated by that position.

Concept systems serve three purposes:

— they expose gaps: a missing subordinate concept within a subdivision signals incomplete terminology;

— they expose overlaps: two concepts sharing the same position in a subdivision signal redundancy;

— they support derivation: definitions, term formation, and entries in terminological collections are drafted from the position and relations of each concept.

4.2 Concepts and characteristics

A concept is constituted by a unique combination of characteristics.

Characteristics are of two kinds:

— **inherent characteristics**, which apply to every instance of the concept and make the concept what it is; and

— **delimiting characteristics** (3.1.12), which distinguish the concept from its coordinate concepts within one subdivision.

In a subdivision, each coordinate concept shall be delimited from the others by at least one delimiting characteristic. Where a coordinate concept has no delimiting characteristic, the subdivision is not MECE (mutually exclusive and collectively exhaustive) and shall be reworked either by refining the criterion of subdivision or by merging the coordinate concepts.

4.3 Concept coordination

Concepts are coordinated — brought into a system — by three canonical coordination methods. Every relation in a concept system is grounded in one of these methods or is associative (5.4).

4.3.1 Generic coordination

Generic coordination subdivides a concept into subordinate concepts that are **kinds** of it: each subordinate concept inherits the intension of the superordinate concept and adds at least one delimiting characteristic.

EXAMPLE In generic coordination, the concept **signal** (information transfer) is subdivided into **analog signal** and **digital signal**, by the delimiting characteristic of value continuity.

© EFS 2026 – All rights reserved

Generic coordination forms the **genus-differentia** structure of definitions: the superordinate concept provides the genus and the delimiting characteristics provide the differentiae.

4.3.2 Partitive coordination

Partitive coordination subdivides a concept into subordinate concepts that are **parts** of it, where the parts fitted together constitute the whole.

Partitive coordination is legitimate only when both conditions hold:

a) the parts, fitted together, constitute the whole; and

b) the parts are coordinate: they share the comprehensive concept (3.1.7) and share a criterion of subdivision.

A partitive subdivision with only a single part is not a partitive coordination; a single part-of assertion is expressed as a binary partitive relation (Table 2, Note).

4.3.3 Instantial coordination

Instantial coordination relates individual concepts to the general concept they realize. An **individual concept** is a concept that corresponds to a unique instance and is not further subdivided by the criteria of terminology work.

EXAMPLE The concept * metre* (the SI unit) is an instance of the general concept **unit of length**; the concept * Protocol for web resource identification* is general, while the concept **the URN namespace `isbn`** is individual.

4.4 Criteria of subdivision and multidimensionality

Every subdivision shall be performed with an explicit criterion of subdivision (3.1.8). The criterion shall be stated in the concept system record and shall remain associated with the subdivision it generated.

Where more than one criterion is applied to the same set of concepts, the concept system is **multidimensional** (3.1.9). Multidimensionality shall be represented as parallel subdivisions, one per criterion, and shall not be represented as a single subdivision whose coordinate concepts are delimited by characteristics drawn from different criteria.

EXAMPLE The concept **road vehicle** may be subdivided by motive (motorized, human-powered) and by environment (on-road, off-road). These are two parallel subdivisions. A single subdivision containing **motorized on-road**, **motorized off-road**, **human-powered on-road**, and **human-powered off-road** collapses two criteria into one level and is not a conforming representation.

NOTE Multidimensionality also applies to n-ary relations: the same comprehensive concept may have several rakes of the same kind, distinguished by their criterion (7.6). Concept management systems shall preserve the distinction between such rakes.

4.5 Concept systems as graphs

A concept system is a directed labelled graph: concepts are nodes, concept relations are labelled edges (binary relations) or labelled hyperedges (n-ary relations). The graph representation makes the following operations well-defined and machine-checkable:

— traversal from any concept to its superordinate, subordinate, and associated concepts;

— validation of the constraints specified in Clause 9;

— projection into vocabularies such as SKOS, where generic coordination projects into broader/narrower hierarchies and associative relations project into skos:related.

Concept systems shall be maintained as directed labelled graphs with explicit labels; concept management systems shall not infer relation types from layout, position, or diagram style alone.

© EFS 2026 – All rights reserved

5 Concept relation vocabulary

5.1 General

The ES relation vocabulary is a controlled vocabulary. Each relation type is defined by:

— its **identifier**, used in data interchange and in registers;

— its **semantics**, stated as the condition under which the relation holds;

— its **direction**, stated in the form “the current concept … the related concept”; and

— its **inverse**, the relation type that holds when the direction is reversed.

Relation types of this vocabulary shall be referenced by their identifiers. Concept management systems shall use the identifiers exactly as given in this clause; systems that require additional relation types shall propose them for registration in the ES relation register (5.6) and shall not extend the vocabulary locally.

Every directed relation type has an inverse in the vocabulary. Where an assertion is recorded, systems should record the inverse explicitly or make it derivable without ambiguity. Symmetric relation types (`opposite_of`, `equivalent`, `exact_match`, `related_match`, `compare`, `related_concept`) have no distinct inverse.

5.2 Hierarchical relations

Hierarchical relations express membership of a subordinate concept in a superordinate concept. The vocabulary distinguishes the three coordination methods of Clause 4.

Table 1 — Table — Generic relation types

Identifier	Semantics	Inverse
`broader_generic`	The current concept is a kind of the related concept.	`narrower_generic`
`narrower_generic`	The related concept is a kind of the current concept.	`broader_generic`
`broader`	The current concept is subsumed by the related concept, coordination method not stated.	`narrower`
`narrower`	The related concept is subsumed by the current concept, coordination method not stated.	`broader`
NOTE The unqualified `broader`/`narrower` pair is provided for interchange with SKOS. Within ES concept systems, the coordination method shall be stated whenever it is known; the unqualified pair is a fallback, not a substitute.		

Table 2 — Table — Partitive relation types (binary)

Identifier	Semantics	Inverse
`broader_partitive`	The current concept is a part of the related concept (whole/part).	`narrower_partitive`
`narrower_partitive`	The related concept is a part of the current concept (whole/part).	`broader_partitive`
`has_part`	The current concept has the related concept as a part.	`is_part_of`
`is_part_of`	The current concept is a part of the related concept.	`has_part`
NOTE Binary partitive types express single part-of assertions without any claim of completeness or per-member presence. Where a decomposition is jointly significant, an n-ary partitive relation is required (7.3).		

Table 3 — Table — Instantial relation types

Identifier	Semantics	Inverse
`broader_instantial`	The current concept is an instance of the related concept.	`narrower_instantial`
`narrower_instantial`	The related concept is an instance of the current concept.	`broader_instantial`
`instance_of`	The current concept is an instance of the related concept.	`has_instance`
`has_instance`	The current concept has the related concept as an instance.	`instance_of`

© EFS 2026 – All rights reserved

5.3 Equivalence and mapping relations

Equivalence relations assert correspondence between concepts. Mapping relations assert correspondence between concepts of **different vocabularies**, in the sense of ISO 25964-2 and SKOS.

Table 4 — Table — Equivalence and mapping relation types

Identifier	Semantics	Inverse
`equivalent`	The current concept is equivalent in denotation to the related concept.	`equivalent` (symmetric)
`exact_match`	The current concept is an exact cross-vocabulary match of the related concept, which can be replaced one-for-one.	`exact_match` (symmetric)
`close_match`	The current concept is approximately equivalent to the related concept.	`close_match` (symmetric)
`broad_match`	The related concept of another vocabulary is broader than the current concept.	`narrow_match`
`narrow_match`	The related concept of another vocabulary is narrower than the current concept.	`broad_match`
`related_match`	The current concept is associatively related to a concept of another vocabulary.	`related_match` (symmetric)
`compare`	The current concept may be compared to the related concept to clarify distinctions.	`compare` (symmetric)
`contrast`	The current concept is understood in contrast to the related concept.	`contrast` (symmetric)
NOTE Cross-language equivalence between localized concepts is governed by EFS 2504 and is recorded as an equivalence degree on the localization, not as a concept relation.		

5.4 Associative relations

Associative relations express thematic, non-hierarchical links. The vocabulary enumerates the associative relation kinds of ISO 704 Table 1 and adds direction-specific identifiers for each kind. Within each following table, the kinds are taken from ISO 704 and each kind is expressed as a directed pair.

Table 5 — Contiguity relation types

Identifier	Semantics	Inverse
`enhances`	The current concept is a tool; the related concept is an accessory that enhances it.	`enhanced_by`
`enhanced_by`	The current concept is an accessory that enhances the related tool.	`enhances`
`contains`	The current concept is a container; the related concept is its content.	`contained_by_container`
`contained_by_container`	The current concept is contained in the related container.	`contains`
`contained_by_contained`	The current concept shares a container with the related concept.	`contained_by_contained` (symmetric)
`attaches`	The current concept is a tool; the related concept is a connection attached to it.	`attached_to`
`attached_to`	The current concept is a connection attached to the related tool.	`attaches`
`made_of`	The current concept is a concrete item; the related concept is the material it is made of.	`material_of`
`material_of`	The current concept is the material of the related item.	`made_of`
`has_property`	The current concept possesses the property expressed by the related concept.	`property_of`
`property_of`	The current concept is a property possessed by the related concept.	`has_property`
`owns`	The current concept owns the related concept.	`owned_by`

© EFS 2026 – All rights reserved

Table 5 *(continued)*

Identifier	Semantics	Inverse
`owned_by`	The current concept is owned by the related concept.	`owns`
`ranks_above`	The current concept outranks the related concept in a hierarchy of rank.	`ranks_below`
`ranks_below`	The current concept is outranked by the related concept.	`ranks_above`

Table 6 — Sequential relation types (binary)

Identifier	Semantics	Inverse
`causes`	The current concept causes the related concept.	`caused_by`
`caused_by`	The current concept is caused by the related concept.	`causes`
`develops_into`	The current concept is an earlier developmental stage of the related concept.	`develops_from`
`develops_from`	The current concept develops from the earlier stage expressed by the related concept.	`develops_into`
`sequentially_related`	The related concept follows or precedes the current concept in an unspecified order of sequence.	`sequentially_related` (symmetric)
`spatially_related`	The current concept is spatially positioned relative to the related concept.	`spatially_related` (symmetric)
`temporally_related`	The current concept is temporally positioned relative to the related concept.	`temporally_related` (symmetric)
NOTE A sequence of two or more jointly significant ordered stages is expressed as an n-ary sequential relation (7.5), not as a chain of binary `causes` or `develops_into` edges.		

Table 7 — Activity relation types

Identifier	Semantics	Inverse
`performed_by`	The current concept is an action; the related concept is its actor.	`performs_action`
`performs_action`	The current concept is the actor of the related action.	`performed_by`
`done_by_method`	The current concept is an action; the related concept is the method used.	`method_of`
`method_of`	The current concept is the method of the related action.	`done_by_method`
`done_at`	The current concept is an action; the related concept is the place where it is done.	`place_of_action`
`place_of_action`	The current concept is the place of the related action.	`done_at`
`done_for_purpose`	The current concept is an action; the related concept is its objective.	`purpose_of_action`
`purpose_of_action`	The current concept is the objective of the related action.	`done_for_purpose`
`done_for_beneficiary`	The current concept is an action; the related concept is its beneficiary.	`beneficiary_of_action`
`beneficiary_of_action`	The current concept is the beneficiary of the related action.	`done_for_beneficiary`
`acts_on_object`	The current concept is an action; the related concept is its object.	`object_of_action`
`object_of_action`	The current concept is the object of the related action.	`acts_on_object`
`done_with_tool`	The current concept is an action; the related concept is the instrument used.	`tool_of_action`
`tool_of_action`	The current concept is the instrument of the related action.	`done_with_tool`
`acts_on_patient`	The current concept is an action; the related concept is the patient undergoing it.	`patient_of_action`
`patient_of_action`	The current concept is the patient of the related action.	`acts_on_patient`

© EFS 2026 – All rights reserved

Table 8 — Instrumental relation types

Identifier	Semantics	Inverse
`produces`	The current concept is the producer; the related concept is the product.	`produced_by`
`produced_by`	The current concept is the product of the related producer.	`produces`
`made_from_ingredient`	The current concept is the product; the related concept is the raw material.	`ingredient_of`
`ingredient_of`	The current concept is the raw material of the related product.	`made_from_ingredient`
`produced_by_instrument`	The current concept is the tool; the related concept is the product produced with it.	`instrument_product_of`
`instrument_product_of`	The current concept is the product produced with the related tool.	`produced_by_instrument`
`employed_by_professional`	The current concept is the professional; the related concept is the tool employed.	`employs_tool`
`employs_tool`	The current concept is the tool employed by the related professional.	`employed_by_professional`
`handled_by_tool`	The current concept is the object; the related concept is the tool used to handle it.	`tool_handles`
`tool_handles`	The current concept is the tool that handles the related object.	`handled_by_tool`
`applied_to_patient`	The current concept is the tool; the related concept is the patient it is applied to.	`patient_of_tool`
`patient_of_tool`	The current concept is the patient to whom the related tool is applied.	`applied_to_patient`

Table 9 — Interactional, origination and transmission relation types

Identifier	Semantics	Inverse
`controls`	The current concept controls the related concept.	`controlled_by`
`controlled_by`	The current concept is controlled by the related concept.	`controls`
`represents`	The current concept represents the related concept.	`represented_by`
`represented_by`	The current concept is represented by the related concept.	`represents`
`originates_from`	The current concept originates from the related concept (producer–product semantics).	`origin_of`
`origin_of`	The current concept is the origin of the related concept.	`originates_from`
`sends_to`	The current concept sends to the related concept.	`receives_from`
`receives_from`	The current concept receives from the related concept.	`sends_to`
`sends_object`	The current concept sends the related object.	`object_sent_by`
`object_sent_by`	The current concept is the object sent by the related sender.	`sends_object`

Table 10 — Opposition relation types

Identifier	Semantics	Inverse
`opposite_of`	The current concept is the opposite of the related concept.	`opposite_of` (symmetric)

5.5 Lexical, lifecycle, register and versioning relations

The following relation types connect concepts to related concepts that are not part of their concept system structure, and manage the lifecycle of concepts in registers.

© EFS 2026 – All rights reserved

Table 11 — Lexical relation types

Identifier	Semantics	Inverse
`see`	The definition of the current concept is completed by referring to the related concept.	`see` (symmetric, informative)
`references`	The current concept references the related concept.	`referenced_by`
`referenced_by`	The current concept is referenced by the related concept.	`references`
`related_concept`	The current concept is associated with the related concept without a stated kind.	`related_concept` (symmetric)
`related_concept_broader`	The current concept is associated with the related concept in a broader sense.	`related_concept_narrower`
`related_concept_narrower`	The current concept is associated with the related concept in a narrower sense.	`related_concept_broader`
`homograph`	The related designation is spelled identically to the current designation but denotes a different concept.	`homograph` (symmetric)
`false_friend`	The related designation looks or sounds similar in another language but denotes a different concept.	`false_friend` (symmetric)

Table 12 — Lifecycle and register relation types (ISO 19135, ISO 10241-1)

Identifier	Semantics	Inverse
`deprecates`	The current concept deprecates the related concept: continued use of the related concept is discouraged.	`deprecated_by`
`deprecated_by`	The current concept has been deprecated by the related concept.	`deprecates`
`supersedes`	The current concept supersedes the related concept and replaces it prospectively.	`superseded_by`
`superseded_by`	The current concept has been superseded by the related concept.	`supersedes`
`replaces`	The current concept replaces the related concept in the register.	`replaced_by`
`replaced_by`	The current concept has been replaced in the register by the related concept.	`replaces`
`invalidates`	The current concept invalidates the related concept.	`invalidated_by`
`invalidated_by`	The current concept has been invalidated by the related concept.	`invalidates`
`retires`	The current concept retires the related concept from the register.	`retired_by`
`retired_by`	The current concept has been retired from the register by the related concept.	`retires`

Table 13 — Structural register relation types (ISO 19135)

Identifier	Semantics	Inverse
`has_concept`	The current register item has the related item as its concept.	`is_concept_of`
`is_concept_of`	The current item is the concept of the related register item.	`has_concept`
`has_definition`	The current concept has the related item as a definition.	`definition_of`
`definition_of`	The current item is a definition of the related concept.	`has_definition`
`inherits`	The current concept inherits characteristics from the related concept.	`inherited_by`
`inherited_by`	The current concept is a source of inheritance for the related concept.	`inherits`

© EFS 2026 – All rights reserved

Table 14 — Versioning relation types

Identifier	Semantics	Inverse
`has_version`	The current concept has the related concept as a version.	`version_of`
`version_of`	The current concept is a version of the related concept.	`has_version`
`current_version`	The related concept is the current version among the versions of the current concept.	`current_version_of`
`current_version_of`	The current concept is the current version of the related concept.	`current_version`

5.6 The relation vocabulary as a registered item

The relation vocabulary of this clause is itself a collection of managed items. ES maintains it in a register operated under the FERIN framework (ISO 19135 procedures), where each relation type is a registered item with a stable identifier, a definition, a registration status, and a version history.

A relation type proposed for use in ES concept systems that is not in this vocabulary shall be submitted to the register as a proposal. Accepted proposals are added to the vocabulary in a new version of this document. Systems shall not use unregistered relation type identifiers in interchange.

INFO: The register is publicly available at the Enosema Standards Registry (https://www.enosema.org/standards/).

6 Binary relations

6.1 When to use a binary relation

A binary relation is the correct assertion when the relation holds between exactly two concepts and neither the joint significance of a group of relations, nor the completeness of a decomposition, nor per-member presence or count is being asserted.

Binary relations express:

— hierarchical membership of one concept under another (5.2);

— pairwise associative links of a stated kind (5.4);

— equivalence and mapping between two concepts (5.3); and

— lifecycle and register assertions between two concepts (5.5).

6.2 Recording direction and inverse

A binary relation assertion shall record:

a) the relation type identifier;

b) the direction, by the ordered pair (current concept, related concept) per the semantics of the type; and

c) optionally, an explanatory content string clarifying the specific assertion.

Where a system materializes inverses, the inverse relation type shall be the one specified in Clause 5. Systems shall not invent inverse identifiers.

6.3 Assertions between designations

Some relation types apply between specific **designations** of concepts rather than between concepts — for example, that one designation is an abbreviated form for another. These designation-level relations use the same vocabulary, restricted to the types whose semantics apply to designations:

© EFS 2026 – All rights reserved

— `abbreviated_form_for`: the current designation is an abbreviated form of the related designation of the same concept;

— `short_form_for`: the current designation is a short form of the related designation of the same concept;

— `full_form_for`: the current designation is the full form of the related abbreviation.

Concept management systems shall record designation-level relations on the designation record and shall not encode them as concept-level relations.

6.4 Informative versus normative relations

Relations in a concept system differ in normative force:

— structural relations (hierarchical, partitive, instantial) are **normative**: they define the intension and extension of concepts and shall be validated for coherence (Clause 9);

— associative relations are **informative** by default: they explain and connect, but do not delimit. A domain can declare an associative kind normative for its purposes (for example, `causes` in a safety domain); such declarations belong to the domain profile, not to this document.

7 N-ary relations

7.1 General

An n-ary relation asserts that a group of concepts is jointly significant with respect to a comprehensive concept. ISO 704 expresses such groups as **rakes** or **brackets** in concept diagrams: a single line drawn from the comprehensive concept to two or more coordinate concepts means that the coordination is joint, not an aggregation of independent pairwise links.

An n-ary relation shall record:

a) the **kind** of the relation: partitive, generic, or sequential;

b) the **comprehensive concept** (3.1.7);

c) the **members**: two or more coordinate concepts, each referencing the comprehensive concept, in order where order is significant;

d) the **completeness** (3.1.10): `complete` or `partial`;

e) the **criterion of subdivision** (7.6) or **criterion of sequencing** (7.5); and

f) optional per-member metadata: presence, count, and delimiting status (7.3).

7.2 The binary-versus-n-ary decision rule

A decomposition shall be recorded as an n-ary relation, not as a set of binary relations, when at least one of the following holds:

a) the membership of the group is **jointly significant**: the subordinate concepts fitted together constitute the comprehensive concept;

b) the **completeness** of the group is asserted (the members exhaust the intension of the comprehensive concept, or explicitly do not);

c) **per-member presence or count** is asserted (some members exist in every instance; some in only some instances; some singly; some multiply); or

d) the **delimiting** part is identified among the members.

© EFS 2026 – All rights reserved

Otherwise — where the assertion is merely that one concept relates to another — a binary relation shall be used.

EXAMPLE 1 A **road vehicle** having an **engine** and a **chassis** is a jointly significant partitive decomposition: a n-ary partitive relation with comprehensive concept `road vehicle` and members `engine` (required, exactly_one) and `chassis` (required, exactly_one) is recorded.

EXAMPLE 2 A **publication** having an **author** is a loose partitive association without joint significance: a binary `has_part`-style relation is recorded, since publications can have any number of authors and the author does not delimit the publication.

7.3 Partitive n-ary relations

A partitive n-ary relation (a **partitive rake**) records the decomposition of a comprehensive concept into the parts that fitted together constitute it.

Per-member metadata follows the ISO 704 diagram conventions:

— **presence** (3.1.11): `required` — the part exists in every instance of the comprehensive concept (solid line); `optional` — the part exists in some instances only (dashed line);

— **count**: `exactly_one` — one instance (single line); `at_least_one` — one or more instances; `multiple` — two or more instances (double line);

— **delimiting status**: where one member carries the delimiting characteristic that distinguishes the comprehensive concept within its own subdivision, it is marked **delimiting** (bold line of three-fold width in ISO 704 diagrams).

EXAMPLE A **railway switch** consists of a **switch blade** (required, exactly_one, delimiting), a **switch motor** (optional, exactly_one) and a **locking device** (required, at_least_one).

The combination of presence and count shall be consistent: an `optional` member shall not be recorded with `exactly_one` count, since presence is already conditional.

7.4 Generic n-ary relations

A generic n-ary relation records the kinds that jointly constitute a comprehensive concept under a stated criterion of subdivision. It is used where the set of kinds is itself significant — for example, to assert that a classification is exhaustive under one criterion:

— the comprehensive concept is the superordinate concept;

— the members are the kinds, which shall be coordinate (3.1.6): they share the comprehensive concept and the criterion;

— `completeness: complete` asserts that the kinds exhaust the extension under the criterion; `completeness: partial` asserts that additional kinds exist or are anticipated.

Generic n-ary relations carry no per-member presence or count metadata; each kind either exists in the system or it does not.

7.5 Sequential n-ary relations

A sequential n-ary relation records two or more subordinate concepts **ordered** by a criterion of sequencing, jointly constituting a comprehensive concept such as a process, a development, or a lifecycle.

The relation shall record:

a) the **criterion of sequencing**: `temporal`, `spatial`, `developmental`, `causal`, `process_step`, or a domain-defined criterion;

b) the members in their order; and

© EFS 2026 – All rights reserved

c) the completeness of the sequence.

EXAMPLE The development of **amphibian** from **larva** to **adult** is a developmental sequence; the stages of a **release lifecycle** (proposed, drafted, reviewed, published) are process steps.

The sequential n-ary relation is preferred over chains of binary `causes`/`develops_into` edges wherever the order and the joint significance of the stages matter; binary sequential edges remain correct for isolated pairwise assertions.

7.6 Criterion records

The criterion of subdivision or sequencing shall be recorded with the n-ary relation, not inferred. A criterion record carries:

a) a human-readable statement of the criterion; and

b) an identifier where the criterion is itself a managed item (for example, a registered subdivision criterion in the domain register).

The same comprehensive concept may carry several n-ary relations of the same kind distinguished only by criterion; systems shall present these as parallel decompositions (Note) and shall not merge them.

NOTE Multidimensionality at the level of concept systems is specified in 4.4; the n-ary criterion record applies the same principle to individual decompositions.

7.7 Registration of n-ary relations

Within a register, an n-ary relation is an item in its own right, identified independently of the concepts it connects:

a) its identifier is stable across member edits;

b) its lifecycle (proposed, accepted, retired) follows the register procedures of ISO 19135 as adopted by the FERIN framework;

c) member edits are recorded as version changes of the relation item, so that citing systems can reference the decomposition as it was at a given time.

Concept management systems shall store n-ary relations as first-class items and shall not reconstruct them by traversing binary edges, since the joint significance, completeness, and per-member metadata are not recoverable from pairwise assertions alone.

7.8 Serialization of n-ary relations

The reference serialization stores each n-ary relation as one file, following the relations-as-files design:

a) the file lives at `relations/<comprehensive-id>/<criterion-slug>.yaml` within the concept dataset, where the comprehensive identifier and the criterion slug match the directory and file names respectively;

b) the relation record carries a stable identifier of the form `<comprehensive-id>/<criterion-slug>`, a **type discriminator** — `partitive_relation`, `generic_relation`, or `sequential_relation` — the comprehensive concept reference, the member list with per-member presence, count and delimiting metadata, the completeness, and the criterion;

c) member array order is insignificant for partitive and generic relations and significant for sequential relations;

d) one comprehensive concept may carry multiple relation files of the same type distinguished by criterion slug, representing parallel decompositions (7.6).

Wire vocabularies shall use the `members` field name uniformly across all n-ary relation types; decompositions shall not serialize members under type-specific field names (such as `partitives`).

© EFS 2026 – All rights reserved

NOTE The glossarist concept model implements this serialization with the three type discriminators; its schema is the reference for the file and identifier forms specified here.

8 Machine-readable representation

8.1 General

Concept relations shall be represented in machine-readable form so that concept systems can be validated, interchanged, and published. This clause specifies the representation requirements; it does not mandate a serialization format, but specifies the semantic content that any conforming serialization shall carry and the vocabulary mappings that interchange formats shall use.

8.2 Semantic content of a relation record

A conforming relation record carries, for a binary relation:

a) the relation type identifier (Clause 5);

b) the identifier of the current (subject) concept;

c) the identifier of the related (object) concept;

d) optionally, an explanatory content string;

e) optionally, references to concept sources (bibliographic support); and

f) optionally, the registration status of the relation assertion itself.

For an n-ary relation, a conforming record additionally carries the kind of the relation, the comprehensive concept identifier, the ordered member list with per-member presence, count and delimiting metadata, the completeness, and the criterion (Clause 7).

8.3 Mapping to SKOS

Where a concept system is published as a SKOS concept scheme, the following mappings shall be used:

Table 15

ES relation type	SKOS property
`broader` / `broader_generic` / `broader_partitive` / `broader_instantial`	`skos:broader`
`narrower` / `narrower_generic` / `narrower_partitive` / `narrower_instantial`	`skos:narrower`
`related_concept` and associative types	`skos:related`
`exact_match`	`skos:exactMatch`
`close_match`	`skos:closeMatch`
`broad_match`	`skos:broadMatch`
`narrow_match`	`skos:narrowMatch`
`related_match`	`skos:relatedMatch`

The projection discards ES-specific information: coordination method, inverse pairing, per-member presence and count, delimiting status, completeness, and criterion are not carried by SKOS. Publishers shall retain the full ES record and treat the SKOS projection as lossy. Systems shall not round-trip ES data through SKOS alone.

8.4 JSON-LD and typed representation

Where concept systems are published as JSON-LD or as typed records:

© EFS 2026 – All rights reserved

— relation types shall be identified by the identifiers of Clause 5, resolvable through the ES relation register (5.6);

— inverses may be materialized but shall remain consistent with the recorded direction;

— n-ary relations shall be represented as first-class resources, not as reified pairs, with members referenced by identifier and per-member metadata attached to the membership, not to the concept.

8.5 Relation identifiers in interchange

Relation type identifiers used in interchange shall be exactly the identifiers of Clause 5 or of a domain profile registered per 5.6. Systems shall reject interchange containing unknown relation type identifiers rather than silently dropping the relations, and shall report them to the sender.

9 Quality constraints on concept systems

9.1 General

A concept system degrades through use: concepts are added without their relations, decompositions are edited without their criteria, and lifecycle events are recorded without their counterparts. This clause specifies the constraints that concept management systems shall enforce to keep a concept system coherent. The constraints are stated so that they are mechanically checkable.

9.2 Structural constraints

The following constraints shall hold for every concept system.

ac1 Every relation type identifier in the system belongs to the vocabulary of Clause 5 or to a registered domain profile (5.6).

ac2 Every binary relation references concepts that exist in the system; dangling references are not permitted.

ac3 Every hierarchical relation is acyclic: a concept shall not be its own ancestor or descendant through any chain of hierarchical relations.

ac4 Every subdivision in the system states its criterion of subdivision; every n-ary relation states its criterion of subdivision or sequencing.

ac5 Every subordinate concept within one subdivision is delimited from its coordinate concepts by at least one delimiting characteristic (Clause 4).

ac6 Where two parallel subdivisions of the same concept exist, they differ in criterion and are recorded as parallel subdivisions (4.4).

9.3 Constraint checking on lifecycle relations

The lifecycle relation types of 5.5 are directional in time. The following constraints apply.

lc1 A concept cannot both `supersede` and `superseded_by` the same target; the inverse, if materialized, shall be consistent with the direction of the assertion.

lc2 A retired concept shall not be the subject of new structural relations, except relations that record the retirement itself.

lc3 Where a concept is replaced in a register, the `replaces`/`replaced_by` pair shall link the register items, and any `supersedes`/`superseded_by` pair shall link the concepts themselves; systems shall not conflate the two pairs.

© EFS 2026 – All rights reserved

9.4 Constraint checking on n-ary relations

na1 Every n-ary relation has at least two members.

na2 Every member of an n-ary relation is coordinate with the other members: they share the comprehensive concept and the criterion.

na3 The presence/count combination of each member is consistent (7.3): `optional` members shall not carry `exactly_one`.

na4 A `complete` n-ary relation whose members are all `optional` is inconsistent: a decomposition of optional members cannot exhaust the intension of the comprehensive concept; systems shall reject the record or require the criterion to be revised.

na5 At most one member of an n-ary relation is marked delimiting.

9.5 Consistency of inverses

For every materialized inverse pair, the system shall verify that the inverse assertion exists and uses the inverse type of Clause 5. Where a system does not materialize inverses, it shall derive them on request from the recorded direction, and the derivation shall use the inverse table of Clause 5.

9.6 Review of concept systems

Concept systems shall be reviewed on a periodic basis (EFS 2513 specifies the organizational review requirements). The review shall, at minimum:

— check the constraints of this clause against the live system;

— report subdivisions whose criterion has been lost or left implicit;

— report lifecycle events recorded without their register counterpart; and

— report SKOS projections that have diverged from the full ES record.

© EFS 2026 – All rights reserved

Bibliography

[1] ISO 639-3[4)] , *Codes for the representation of names of languages — Part 3: Alpha-3 code for comprehensive coverage of languages*.

[2] ISO 10241-2, *Terminological entries in standards — Part 2: Adoption of standardized terminological entries*.

[3] ISO 15924, *Information and documentation — Codes for the representation of names of scripts*.

[4] ISO 24229, *Information and documentation — Codes for written language conversion systems*.

[5] FERIN, *Framework for Extensible Registration of INformation*, https://www.ferin.org/

[6] EFS 2501:2025, *Terminology management — Principles and methods*

[7] EFS 2504:2025, *Terminology management — Multilingual terminology*

[8] EFS 2507:2025, *Terminology management — Concept relationships*

[9] EFS 2513:2026, *Terminology management — Concept management systems — Requirements*

4) Withdrawn.

© EFS 2026 – All rights reserved

enosema

Price based on 18 pages

© EFS 2026
All rights reserved

Making meaning meaningful