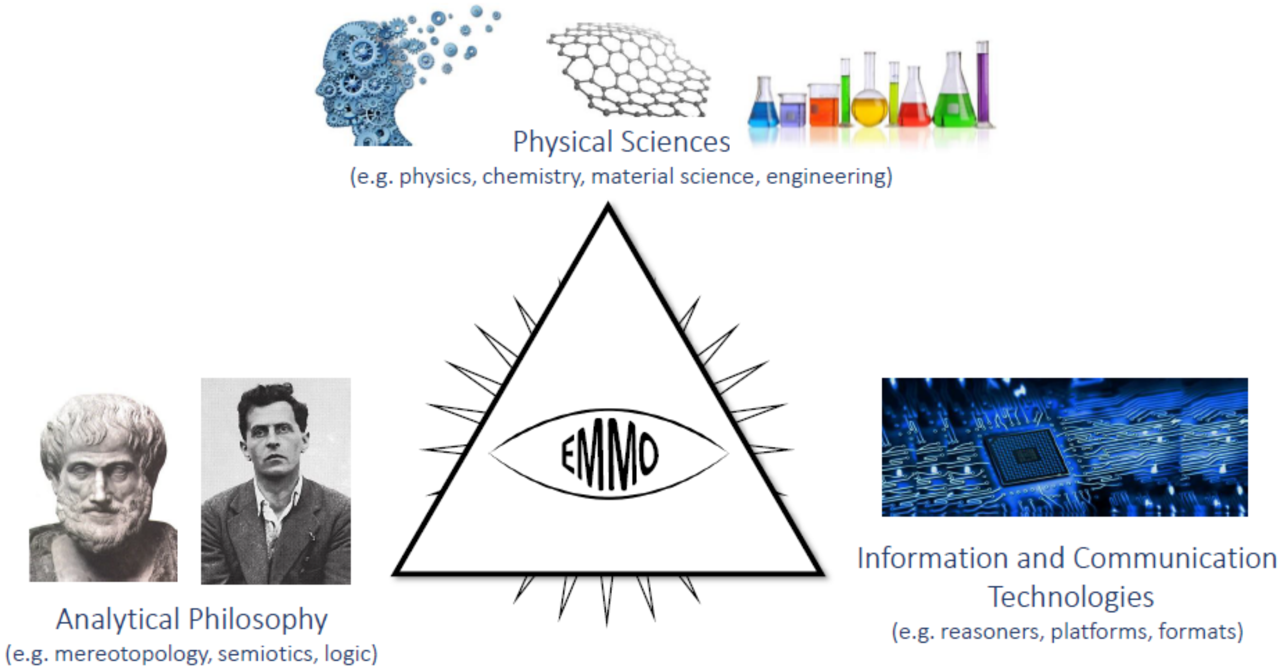
European Materials Modelling Ontology

Version 1.0.0-alpha

European Materials Modelling Counsil (EMMC)

March 18, 2020



Abstract

EMMO is an ontology that is created by the Europeean Materials Modelling Council (EMMC) to provide a formal way to describe the fundamental concepts of physics, chemistry and materials science. EMMO is designed to pave the road for semantic interoperability providing a generic common ground for describing materials, models and data that can be adapted by all domains.

It is a representational framework of predefined classes and axioms (ontology) provided by experts (EMMC) that enables end users (industry, research, academy) to represent real life physical entities (materials, devices), models and properties using ontological signs (individuals) in a standard way to facilitate interactions and exchanges (data, software, knowledge) between all involved material modelling and characterization communities and stakeholders.

Keywords: EMMO, materials science, modelling, characterisation, materials, ontology

Authors:

Emanuele Ghedini, University of Bologne

Gerhard Goldbeck, Goldbeck Consulting

Jesper Friis, SINTEF

Adham Hashibon, Fraunhofer IWM

Georg Schmitz, ACCESS

Contents

Chapter 1

Introduction

EMMO is a multidisciplinary effort to develop a standard representational framework (the ontology) based on current materials modelling knowledge, including physical sciences, analytical philosophy and information and communication technologies. This multidisciplinarity is illustrated by the figure on the title page. It provides the connection between the physical world, materials characterisation world and materials modelling world.

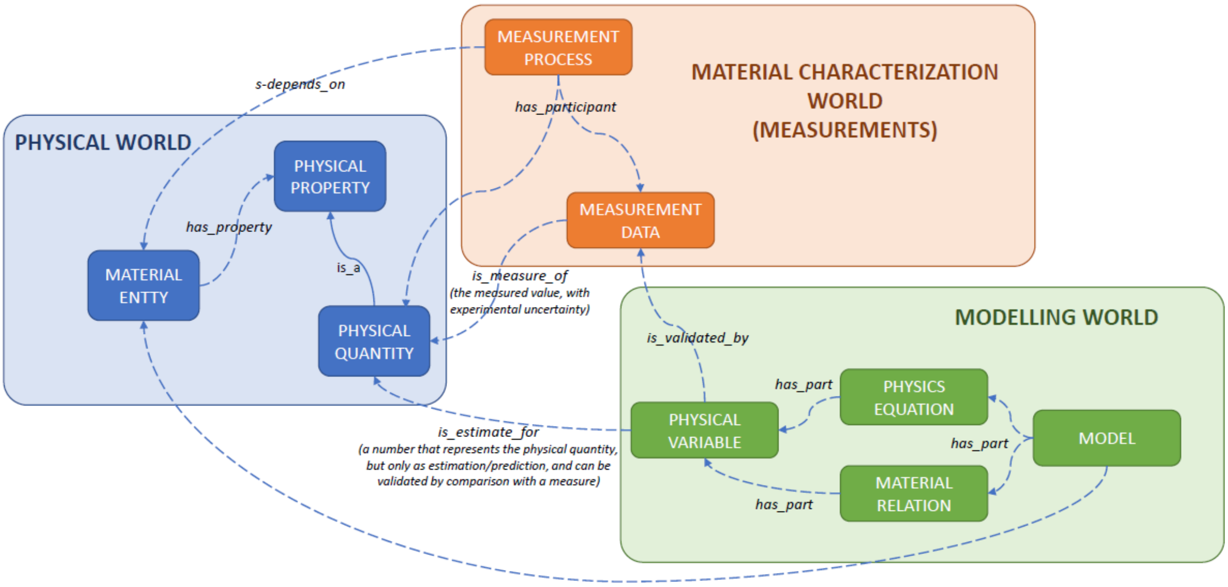


Figure 1.1: EMMO provides the connection between the physical world, materials characterisation world and materials modelling world.

EMMO is based on and is consistent with the Review of Materials Modelling, CEN Workshop Agreement and MODA template. However, while these efforts are written for humans, EMMO is defined using the Web Ontology Language (OWL), which is machine readable and allows for machine reasoning. In terms of semantic representation, EMMO brings everything to a much higher level than these foundations.

As illustrated in the figure below, EMMO covers all aspects of materials modelling and characterisation, including:

- the **material** itself, which must be described in a rigorous way
- the **observation process** involving an observer that percieves the real world (characterisation)
- the **properties** that are measured or modelled
- the **physics laws** that describe the material behaviour
- the **physical models** that approximate the physics laws
- the **solver** including the numerical discretisation method that leads to a solvable mathematical representation under certain simplifying assumptions
- the **numerical solver** that performs the calculations
- the **post processing** of experimental or simulated data

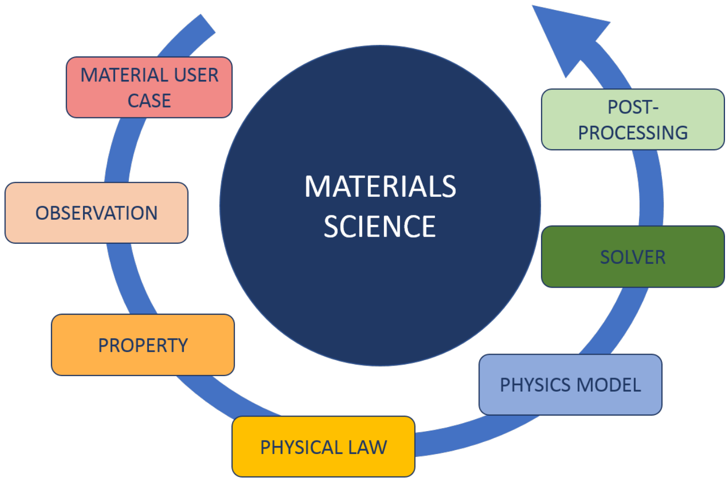


Figure 1.2: The aspects of materials modelling and characterisation covered by EMMO.

EMMO is released under the Creative Commons license and is available at emmo.info/. The OWL2-DL sources are available in RDF/XML format.

What is an ontology

In short, an ontology is a specification of a conceptualization. The word *ontology* has a long history in philosophy, in which it refers to the subject of existence. The so-called ontological argument for the existence of God was proposed by Anselm of Canterbury in 1078. He defined God as *"that than which nothing greater can be thought"*, and argued that *"if the greatest possible being exists in the mind, it must also exist in reality. If it only exists in the mind, then an even greater being must be possible – one which exists both in the mind and in reality"*. Even though this example has little to do with todays use of ontologies in e.g. computer science, it illustrates the basic idea; the ontology defines some basic premises (concepts and relations between them) from which it is possible reason to gain new knowledge.

For a more elaborated and modern definition of the ontology we refer the reader to the one provided by Tom Gruber (2009). Another useful introduction to ontologies is the paper Ontology Development 101: A Guide to Creating Your First Ontology by Noy and McGuinness (2001), which is based on the Protege sortware, with which EMMO has been developed.

A taxonomy is a hierarchical representation of classes and subclasses connected via `is_a` relations. Hence, it is a subset of the ontology excluding all but the `is_a` relations. The main use of taxonomies is for the organisation of classifications. The figure shows a simple example of a taxonomy illustrating a categorisation of four classes into a hierarchy of more higher of levels of generality.

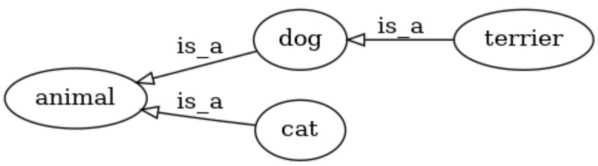


Figure 1.3: Example of a taxonomy.

In EMMO, the taxonomy is a rooted directed acyclic graph (DAG). This is important since many classification methods relies on this property, see e.g. Valentini (2014) and Robison et al (2015). Note, that EMMO is a DAG does not prevent some classes from having more than one parent. A `Variable` is for instance both a `Mathematical` and a `Symbol`. See appendix for the full EMMO taxonomy.

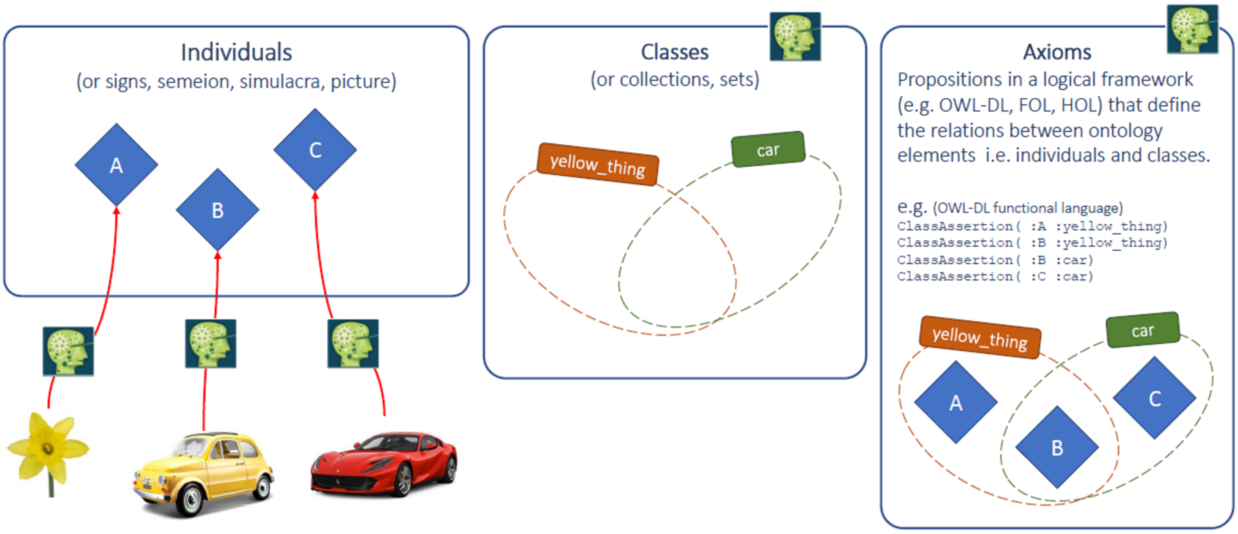


Figure 1.4: The primitive building blocks of EMMO.

Primitive elements in EMMO

Individuals

Individuals are the basic, "ground level" components of EMMO. They may include concrete objects such as cars, flowers, stars, persons and molecules, as well as abstract individuals such as a measured height, a specific equation and software programs.

Individuals possess attributes in form of axioms that are defined by the user (interpreter) upon declaration.

Classes

Classes represent concepts. They are the building blocks that we use to create an ontology as a representation of knowledge. We distinguish between *defined* and *non-defined* classes.

Defined classes are defined by the requirements for being a member of the class. In the graphical representations of EMMO, defined classes are orange. For instance, in the graph of the top-level entity branch below, The root `EMMO` and a defined class (defined to be the disjoint union of `Item` and `Collection`).

Non-defined classes are defined as an abstract group of objects, whose members are defined as belonging to the class. They are yellow in the graphical representations.

Axioms

Axioms are propositions in a logical framework that define the relations between the individuals and classes. They are used to categorise individuals in classes and to define the *defined* classes.

The simplest form of a class axiom is a class description that just states the existence of the class and gives it an unique identifier. In order to provide more knowledge about the class, class axioms typically contain additional components that state necessary and/or sufficient characteristics of the class. OWL contains three language constructs for combining class descriptions into class axioms:

- *Subclass* (`rdfs:subClassOf`) allows one to say that the class extension of a class description is a subset of the class extension of another class description.
- *Equivalence* (`owl:equivalentClass`) allows one to say that a class description has exactly the same class extension (i.e. the individuals associated with the class) as another class description.
- *Distjointness* (`owl:disjointWith`) allows one to say that the class extension of a class description has no members in common with the class extension of another class description.

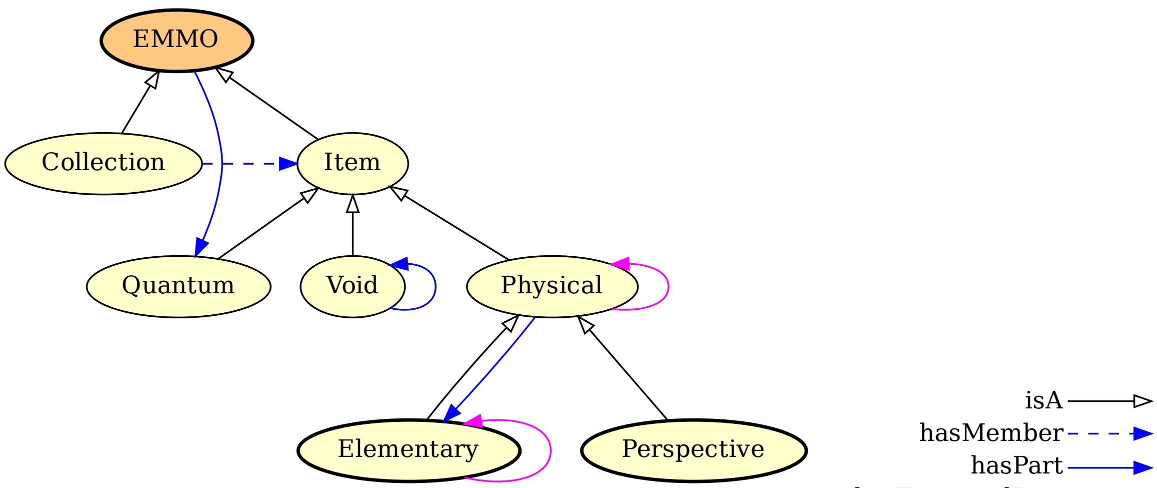


Figure 1.5: Example of the top-level branch of EMMO showing some classes and relationships between them.

See the section about Description logic for more information about these language constructs. Axioms are also used to define relations between relations. These are further detailed in the chapter on Relations.

Theoretical foundations

EMMO build upon several theoretical frameworks.

Semiotics

Semiotics is the study of meaning-making. It is the dicipline of formulating something that possibly can exist in a defined space and time in the real world.

Mereotopology

Mereotopology is the combination of **mereology** (science of parthood) and **topology** (mathematical study of the geometrical properties and conservation through deformations). It is introdused via the `Item` class and based on the `mereotopological` relations. Items in EMMO are always topologically connected in space and time. EMMO makes a strong distinction between membership and parthood relations. In contrast to collections, items can only have parts that are themselves items. For further information, see Casati and Varzi "Parts and Places" (1999).

Physics

EMMO is strongly based on physics, with the aim of being able to describe all aspects and all domains of physics, from quantum mechanics to continuum, engeneering, chemistry, etc. EMMO is compatible with both the De Broglie - Bohm and the Copenhagen interpretation of quantum mecanics (see `Physical` for more comments).

EMMO defines a physics-based parthood hierachy under `Physical` by introducing the following concepts (illustrated in the figure below):

- `Elementary` is the fundamental, non-divisible constituent of entities. In EMMO, elementaries are based on the standard model of physics.

- State is a Physical whose parts does not change during its life time (at the chosen level of granularity). This is consistent with a state within e.g. thermodynamics.
- Existent is a succession of states.

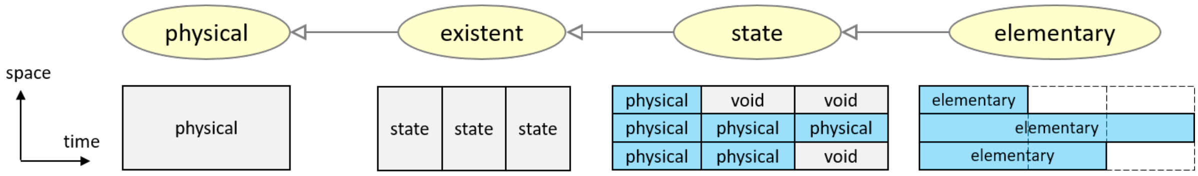


Figure 1.6: Parthood hierachy under Physical.

Metrology

Metrology is the science of measurements. It introduces units and links them to properties. The description of metrology in EMMO is based on the standards of International System of Quantities (ISQ) and International System of Units (SI).

Description logic

Description logic (DL) is a formal knowledge representation language in which the *axioms* are expressed. It is less expressive than first-order logic (FOL), but commonly used for providing the logical formalism for ontologies and semantic web. EMMO is expressed in the Web Ontology Language (OWL), which in turn is based on DL. This brings along features like reasoning.

Since it is essential to have a basic notion of OWL and DL, we include here a very brief overview. For a proper introduction to OWL and DL, we refer the reader to sources like Grau et.al. (2008), OWL2 Primer and OWL Reference.

OWL distinguishes between six types of class descriptions:

1. a class identifier (a IRI reference)
2. an exhaustive enumeration of individuals that together form the instances of a class (`owl:oneOf`)
3. a property restriction (`owl:someValuesFrom`, `owl:allValuesFrom`, `owl:hasValue`, `owl:cardinality`, `owl:minCardinality`, `owl:maxCardinality`)
4. the intersection of two or more class descriptions (`owl:intersectionOf`)
5. the union of two or more class descriptions (`owl:unionOf`)
6. the complement of a class description (`owl:complementOf`)

Except for the first, all of these refer to *defined classes*. The table below shows the notation in OWL, DL and the Manchester OWL syntax, all commonly used for the definitions. The Manchester syntax is used by Protege and is designed to not use DL symbols and to be easy and quick to read and write. Several other syntaxes exist for DL. An interesting example is the pure Python syntax proposed by Lamy (2017), which is used in the open source Owlready2 Python package. The Python API for EMMO is also based on Owlready2.

Table 1.1: Notation for DL and Protege. A and B are classes, R is an active relation, S is an passive relation, a and b are individuals and n is a literal. Inspired by the Great table of Description Logics.

DL	Manchester	Python + Owlready2	Read	Meaning
Constants				
$\top$		Thing	top	A special class with every individual as an instance
$\bot$		Nothing	bottom	The empty class
Axioms				
$A \doteq B$			A is defined to be equal to B	Class *definition*

DL	Manchester	Python + Owlready2	Read	Meaning
$A \sqsubseteq B$	A subclass_of B	class A(B): . . .	all A are B	Class *inclusion*
		issubclass(A, B)		Test for *inclusion*
$A \equiv B$	A equivalent_to B	A.equivalent_to.append(B)	A is equivalent to B	Class *equivalence*
		B in A.equivalent_to		Test for equivalence
$a : A$	a is_a A	a = A()	a is a A	Class *assertion* (*instantiation*)
		isinstance(a, A)		Test for instance of
$(a, b) : R$	a object property assertion b	a.R.append(b)	a is R-related to b	Property *assertion*
$(a, n) : R$	a data property assertion n	a.R.append(n)	a is R-related to n	Data *assertion*
Constructions				
$A \sqcap B$	A and B	A & B	A and B	Class *intersection* (*conjunction*)
$A \sqcup B$	A or B	A | B	A or B	Class *union* (*disjunction*)
$\neg A$	not A	Not(A)	not A	Class *complement* (*negation*)
$\{a, b, ...\}$	{a, b, . . . }	OneOf([a, b, . . .])	one of a, b, . . .	Class *enumeration*
$S \equiv R^{-}$	S inverse_of R	Inverse(R)	S is inverse of R	Property *inverse*
		S.inverse == R		Test for *inverse*
$\forall R.A$	R only A	R.only(A)	all A with R	*Universal restriction*
$\exists R.A$	R some A	R.some(A)	some A with R	*Existential restriction*
$= nR.A$	R exactly n A	R.exactly(n, A)		*Cardinality restriction*
$\leq nR.A$	R min n A	R.min(n, A)		*Minimum cardinality restriction*
$\geq nR.A$	R max n A	R.max(n, A)		*Minimum cardinality restriction*
$\exists R\{a\}$	R value a	R.value(a)		*Value restriction*
Decompositions				
$A \sqcup B \sqsubseteq \bot$	A disjoint with B	AllDisjoint([A, B])	A disjoint with B	Disjoint
		B in A.disjoints()		Test for disjointness
$\exists R.\top \sqsubseteq A$	R domain A	R.domain = A		Classes that the restriction applies to
$\top \sqsubseteq \forall R.B$	R range B	R.range = B		All classes that can be the value of the restriction

Examples

Here are some examples of different class descriptions using both the DL and Manchester notation.

Equivalence (`owl:equivalentTo`)

Equivalence ($\equiv$) defines necessary and sufficient conditions.

Parent is equivalent to mother or father

DL: parent $\equiv$ mother $\vee$ father

Manchester: `parent equivalent_to mother or father`

Inclusion (`rdf:subclassOf`)

Inclusion ($\sqsubseteq$) defines necessary conditions.

An employee is a person.

DL: employee $\sqsubseteq$ person

Manchester: `employee is_a person`

Enumeration (`owl:oneOf`)

The color of a wine is either white, rose or red:

DL: wine_color $\equiv$ {white, rose, red}

Manchester: `wine_color equivalent_to {white, rose, red}`

Existential restriction (`owl:someValuesFrom`)

A mother is a woman that has a child (some person):

DL: mother $\equiv$ woman $\sqcap$ $\exists$has_child.person

Manchester: `mother equivalent_to woman and has_child some person`

Universal restriction (`owl:allValuesFrom`)

All parents that only have daughters:

DL: parents_with_only_daughters $\equiv$ person $\sqcap$ $\forall$has_child.woman

Manchester: `parents_with_only_daughters equivalent_to person and has_child only woman`

Value restriction (`owl:hasValue`)

The owl:hasValue restriction allows to define classes based on the existence of particular property values. There must be at least one matching property value.

All children of Mary:

DL: Marys_children $\equiv$ person $\sqcap$ $\exists$has_parent.{Mary}

Manchester: `Marys_children equivalent_to person and has_parent value Mary`

Property cardinality (`owl:cardinality`)

The owl:cardinality restrictions ($\geq$, $\leq$ or $\equiv$) allow to define classes based on the maximum (owl:maxCardinality), minimum (owl:minCardinality) or exact (owl:cardinality) number of occurences.

A person with one parent:

DL: half_orphant $\equiv$ person and =1has_parent.person

Manchester: `half_orphant equivalent_to person and has_parent exactly 1 person`

Intersection (`owl:intersectionOf`)

Individuals of the intersection ($\sqcap$) of two classes, are simultaneously instances of both classes.

A man is a person that is male:

DL: man $\equiv$ person $\sqcap$ male

Manchester: `man equivalent_to person and male`

Union (`owl:unionOf`)

Individuals of the union (⊔) of two classes, are either instances of one or both classes.

A person is a man or woman:

DL: `person` ≡ `man` ⊔ `woman`

Manchester: `person equivalent_to man or woman`

Complement (`owl:complementOf`)

Individuals of the complement (¬) of a class, are all individuals that are not member of the class.

Not a man:

DL: `female` ≡ ¬ `male`

Manchester: `female equivalent_to not male`

The structure of EMMO

The EMMO ontology is structured in shells, expressed by specific ontology fragments, that extends from fundamental concepts to the application domains, following the dependency flow.

Top Level

The EMMO top level is the group of fundamental axioms that constitute the philosophical foundation of the EMMO. Adopting a physicalistic/nominalistic perspective, the EMMO defines real world objects as 4D objects that are always extended in space and time (i.e. real world objects cannot be spaceless nor timeless). For this reason abstract objects, i.e. objects that does not extend in space and time, are forbidden in the EMMO.

EMMO is strongly based on the analytical philosophy dicipline semiotic. The role of abstract objects are in EMMO fulfilled by semiotic objects, i.e. real world objects (e.g. symbol or sign) that stand for other real world objects that are to be interpreted by an agent. These symbols appear in actions (semiotic processes) meant to communicate meaning by establishing relationships between symbols (signs).

Another important building block of from analytical philosophy is atomistic mereology applied to 4D objects. The EMMO calls it ‘quantum mereology’, since the there is a epistemological limit to how fine we can resolve space and time due to the uncertanity principles.

The mereotopology module introduces the fundamental mereotopological concepts and their relations with the real world objects that they represent. The EMMO uses mereotopology as the ground for all the subsequent ontology modules. The concept of topological connection is used to define the first distinction between ontology entities namely the *Item* and *Collection* classes. Items are causally self-connected objects, while collections are causally disconnected. Quantum mereology is represented by the *Quantum* class. This module introduces also the fundamental mereotopological relations used to distinguish between space and time dimensions.

The physical module, defines the *Physical* objects and the concept of *Void* that plays a fundamental role in the description of multiscale objects and quantum systems. It also define the *Elementary* class, that restricts mereological atomism in space.

In EMMO, the only univocally defined real world object is the *Item* individual called **Universe** that stands for the universe. Every other real world object is a composition of elementaries up to the most comprehensive object; the **Universe**. Intermediate objects are not univocally defined, but their definition is provided according to some specific philosophical perspectives. This is an expression of reductionism (i.e. objects are made of sub-objects) and epistemological pluralism (i.e. objects are always defined according to the perspective of an interpreter, or a class of interpreters).

The *Perspective* class collects the different ways to represent the objects that populate the conceptual region between the elementary and universe levels.

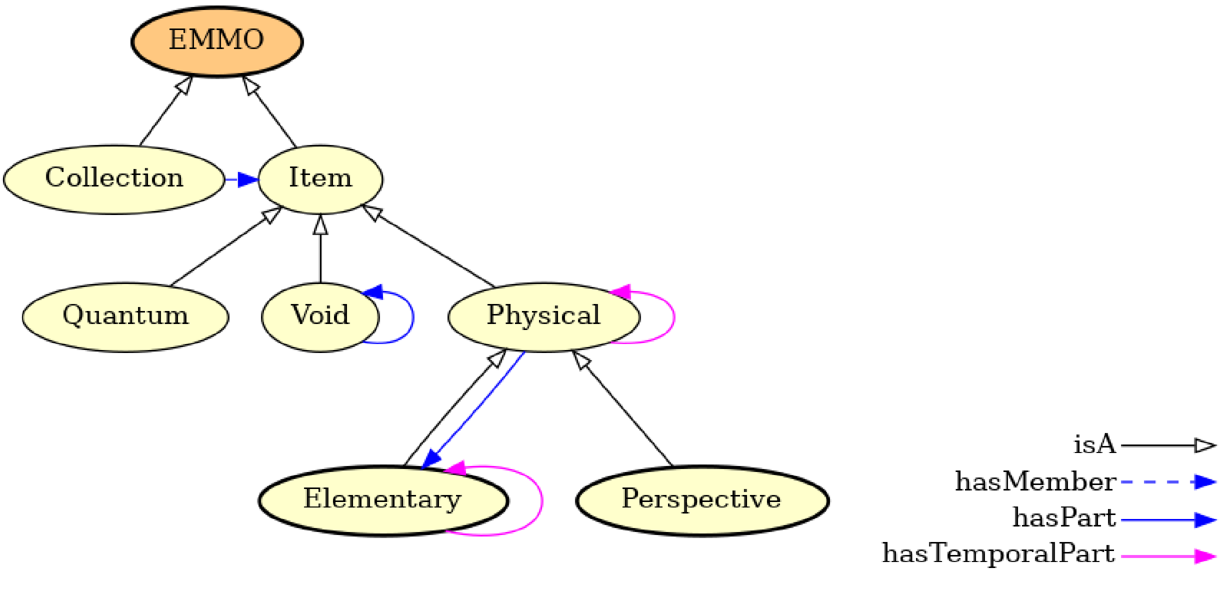


Figure 1.7: The EMMO top level.

Middle Level

The middle level ontologies act as roots for extending the EMMO towards specific application domains.

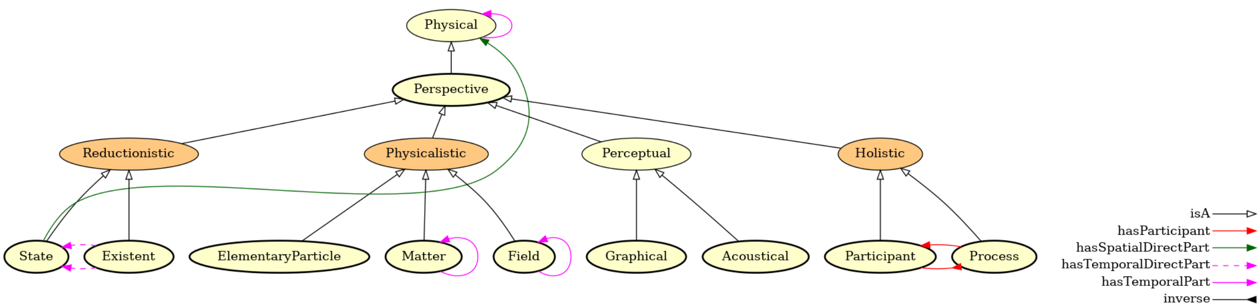


Figure 1.8: The EMMO perspectives.

The *Reductionistic* perspective class uses the fundamental non-transitive parthood relation, called direct parthood, to provide a powerful granularity description of multiscale real world objects. The EMMO can in principle represents the **Universe** with direct parthood relations as a direct rooted tree up to its elementary constituents.

The *Phenomenic* perspective class introduces the concept of real world objects that express of a recognisable pattern in space or time that impress the user. Under this class the EMMO categorises e.g. formal languages, pictures, geometry, mathematics and sounds. Phenomenic objects can be used in a semiotic process as signs.

The *Physicalistic* perspective class introduces the concept of real world objects that have a meaning for the under applied physics perspective.

The *Holistic* perspective class introduces the concept of real world objects that unfold in time in a way that has a meaning for the EMMO user, through the definition of the classes *Process* and *Participant*. The semiotics module introduces the concepts of semiotics and the *Semiosis* process that has a *Sign*, an *Object* and an *Interpreter* as participants. This forms the basis in EMMO to represent e.g. models, formal languages, theories, information and properties.

EMMO relations

All EMMO relations are subrelations of the relations found in the two roots: *mereotopological* and *semiotical*. The relation hierarchy extends more vertically (i.e. more subrelations) than horizontally (i.e. less sibling relations), facilitating the categorisation and inferencing of individuals. See also the chapter EMMO Relations.

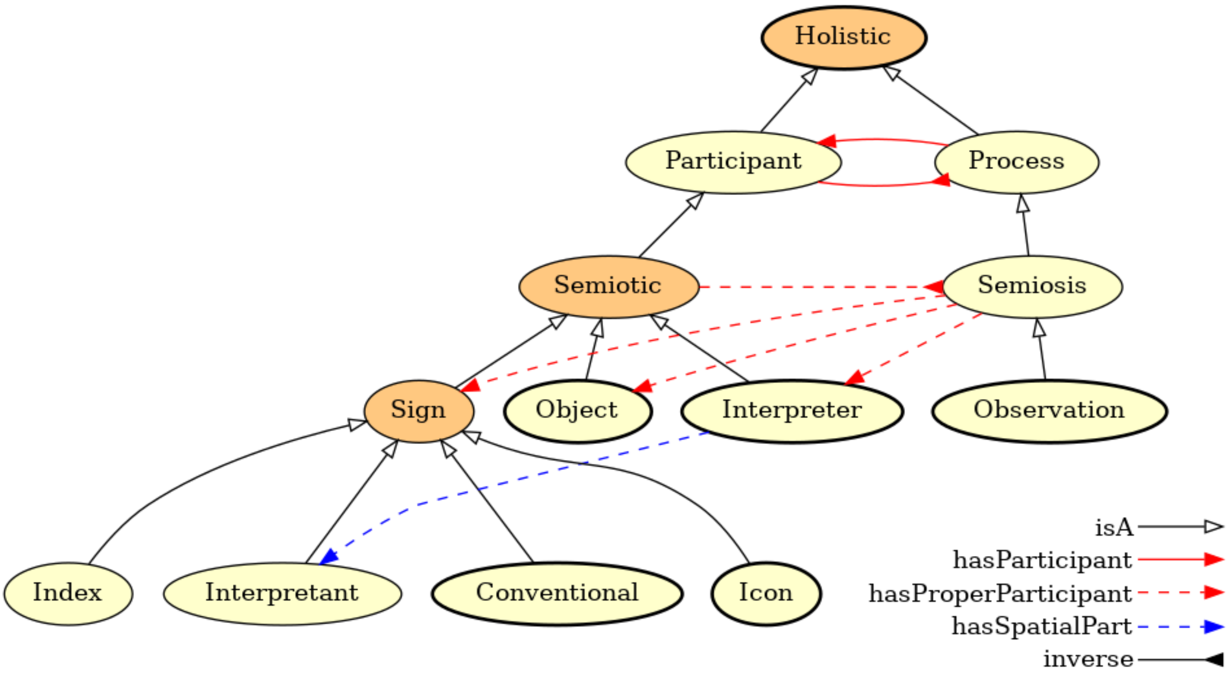


Figure 1.9: The semiotic level, showing both the taxonomy (open black arrows) and other relations as listed in the caption. The inverted arrows corresponds to inverse relations.

Imposing all relations to fall under mereotopology or semiotics is how the EMMO force the developers to respect its perspectives. Two entities are related only by contact or parthood (mereotopology) or by standing one for another (semiosis): no other types of relation are possible within the EMMO.

A unique feature in EMMO, is the introduction of *direct parthood*. As illustrated in the figure below, it is a mereological relation that lacks transitivity. This makes it possible to entities made of parts at different levels of granularity and to go between granularity levels in a well-defined manner. This is paramount for cross scale interoperability. Every material in EMMO is placed on a granularity level and the ontology gives information about the direct upper and direct lower level classes using the non-transitive direct parthood relations.

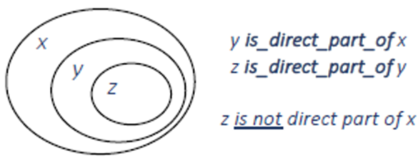


Figure 1.10: Direct parthood.

Annotations

All entities and relations in EMMO have some attributes, called *annotations*. In some cases, only the required *International Resource Identifier* (IRI) and *relations* are provided. However, descriptive annotations, like *elucidation* and *comment*, are planned to be added for all classes and relations. Possible annotations are:

- **Elucidation** is a human readable explanation and clearification of the documented class or relation.
- **Example** clearifies the elucidation through an example. A class may have several examples, each addressing different aspects.
- **Comment** is a clearifying note complementing the definition and elucidation. A class may have several comments, each clearifying different aspects.
- **IRI** stands for *international resource identifier*. It is an identifier that uniquely identifies the class or relation. IRIs are similar to URIs, but are not restricted to the ASCII character set. In EMMO, the IRIs are now valid URLs pointing to the stable version of EMMO.
- **Relations** is a list of relations applying to the current class or relation. The relations for relations are special and will be elaborated on in the introduction to chapter [Relations]. Some of the listed relations are

defined in the OWL sources, while other are inferred by the reasoner. The relations are expressed using the Manchester OWL syntax introduced in section Description logic.

Chapter 2

EMMO Relations

In the language of OWL, relations are called *properties*. However, since relations describe relations between classes and individuals and since properties has an other meaning in EMMO, we only call them *relations*.

Resource Description Framework (RDF) is a W3C standard that is widely used for describing informations on the web and is one of the standards that OWL builds on. RDF expresses information in form of *subject-predicate-object* triplets. The subject and object are resources (aka items to describe) and the predicate expresses a relationship between the subject and the object.

In OWL are the subject and object classes or individuals (or data) while the predicate is a relation. An example of an relationship is the statement *dog is_a animal*. Here `dog` is the subject, `is_a` the predicate and `animal` the object.

OWL distingues between *object properties*, that link classes or individuals to classes or individuals, and *data properties* that link individuals to data values. Since EMMO only deals with classes, we will only be discussing object properties. However, in actual simulation or characterisation applications build on EMMO, datatype propertyes will be important.

The characteristics of the different properties are described by the following *property axioms*:

- `rdf:subPropertyOf` is used to define that a property is a subproperty of some other property. For instance, in the figure below showing the relation branch, we see that `active_relation` is a subproperty or `relation`. The `rdf:subPropertyOf` axioms forms a taxonomy-like tree for relations.
- `owl:equivalentProperty` states that two properties have the same property extension.
- `owl:inverseOf` axioms relate active relations to their corresponding passive relations, and vice versa. The root relation `relation` is its own inverse.
- `owl:FunctionalProperty` is a property that can have only one (unique) value y for each instance x, i.e. there cannot be two distinct values y1 and y2 such that the pairs (x,y1) and (x,y2) are both instances of this property. Both object properties and datatype properties can be declared as "functional".
- `owl:InverseFunctionalProperty`
- `owl:TransitiveProperty` states that if a pair (x,y) is an instance of P, and the pair (y,z) is instance of P, then we can infer that the pair (x,z) is also an instance of P.
- `owl:SymmetricProperty` states that if the pair (x,y) is an instance of P, then the pair (y,x) is also an instance of P. A popular example of a symmetric property is the `siblingOf` relation.
- `rdfs:domain` specifies which classes the property applies to. Or said differently, the valid values of the *subject* in a *subject-predicate-object* triplet.
- `rdfs:range` specifies the property extension, i.e. the valid values of the *object* in a *subject-predicate-object* triplet.

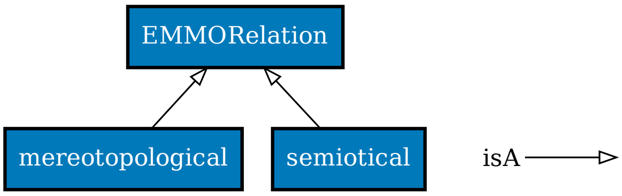


Figure 2.1: Top-level of the EMMO relation hierarchy.

Root of EMMO relations

EMMORelation

IRI: http://emmo.info/emmo/top/mereotopology#EMMO_ec2472ae_cf4a_46a5_8555_1556f5a6c3c5

Elucidation: The superclass of all relations used by the EMMO.

Relations:

- is_a owl:ObjectProperty
- is_a owl:SymmetricProperty
- is_a owl:TransitiveProperty
- is_a owl:topObjectProperty
- inverse_of EMMORelation
- domain EMMO
- range EMMO

Mereotopological branch

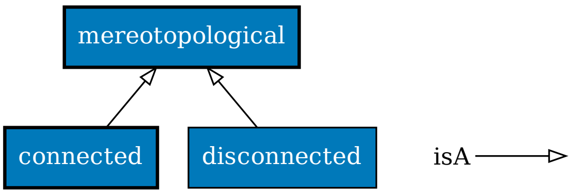


Figure 2.2: Mereotopological branch.

mereotopological

IRI: http://emmo.info/emmo/top/mereotopology#EMMO_03212fd7_abfd_4828_9c8e_62c293052d4b

Elucidation: The superclass of all EMMO mereotopological relations.

Comment: Mereotopology merges mereological and topological concepts and provides relations between wholes, parts, boundaries, etc.

Relations:

- is_a owl:ObjectProperty
- is_a owl:SymmetricProperty
- is_a owl:TransitiveProperty

- is_a EMMORelation
- Inverse(mereotopology.EMMORelation)
- inverse_of mereotopological

disconnected

IRI: http://emmo.info/emmo/top/mereotopology#EMMO_517dfaf9_4970_41ac_81ee_d031627d2c7c

Relations:

- is_a owl:ObjectProperty
- is_a owl:SymmetricProperty
- is_a mereotopological
- Inverse(mereotopology.mereotopological)
- inverse_of disconnected

Connected branch

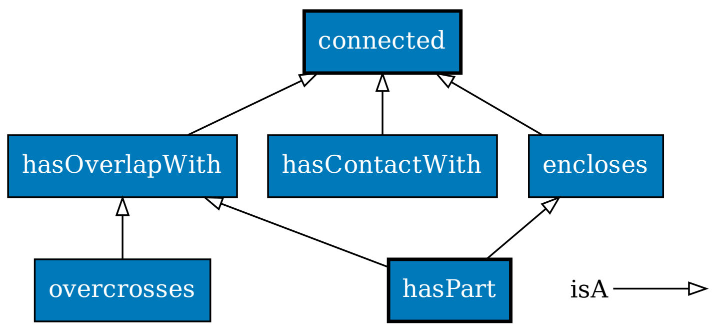


Figure 2.3: Connected branch.

overcrosses

IRI: http://emmo.info/emmo/top/mereotopology#EMMO_9cb984ca_48ad_4864_b09e_50d3fff19420

Relations:

- is_a owl:ObjectProperty
- is_a owl:SymmetricProperty
- is_a hasOverlapWith
- Inverse(mereotopology.hasOverlapWith)
- inverse_of overcrosses

hasContactWith

IRI: http://emmo.info/emmo/top/mereotopology#EMMO_4d6504f1_c470_4ce9_b941_bbbebc9ab05d

Relations:

- is_a owl:ObjectProperty
- is_a owl:SymmetricProperty
- is_a connected
- Inverse(mereotopology.connected)
- inverse_of hasContactWith

encloses

IRI: http://emmo.info/emmo/top/mereotopology#EMMO_8c898653_1118_4682_9bbf_6cc334d16a99

Comment: Enclosure is reflexive and transitive.

Relations:

- is_a owl:ObjectProperty
- is_a owl:TransitiveProperty
- is_a connected
- Inverse(mereotopology.connected)

connected

IRI: http://emmo.info/emmo/top/mereotopology#EMMO_6703954e_34c4_4a15_a9e7_f313760ae1a8

Comment: Causality is a topological property between connected items.

Comment: Items being connected means that there is a topological contact or "interaction" between them.

Relations:

- is_a owl:ObjectProperty
- is_a owl:SymmetricProperty
- is_a mereotopological
- Inverse(mereotopology.mereotopological)
- inverse_of connected

hasOverlapWith

IRI: http://emmo.info/emmo/top/mereotopology#EMMO_d893d373_b579_4867_841e_1c2b31a8d2c6

Relations:

- is_a owl:ObjectProperty
- is_a owl:SymmetricProperty
- is_a connected
- Inverse(mereotopology.connected)
- inverse_of hasOverlapWith

Has Part branch

hasProperParticipant

IRI: http://emmo.info/emmo/middle/holistic#EMMO_c5aae418_1622_4d02_93c5_21159e28e6c1

Relations:

- is_a owl:ObjectProperty
- is_a hasParticipant
- is_a hasProperPart

hasTemporalDirectPart

IRI: http://emmo.info/emmo/middle/reductionistic#EMMO_65a2c5b8_e4d8_4a51_b2f8_e55effc0547d

Relations:

- is_a owl:ObjectProperty
- is_a owl:InverseFunctionalProperty
- is_a owl:AsymmetricProperty
- is_a owl:IrreflexiveProperty

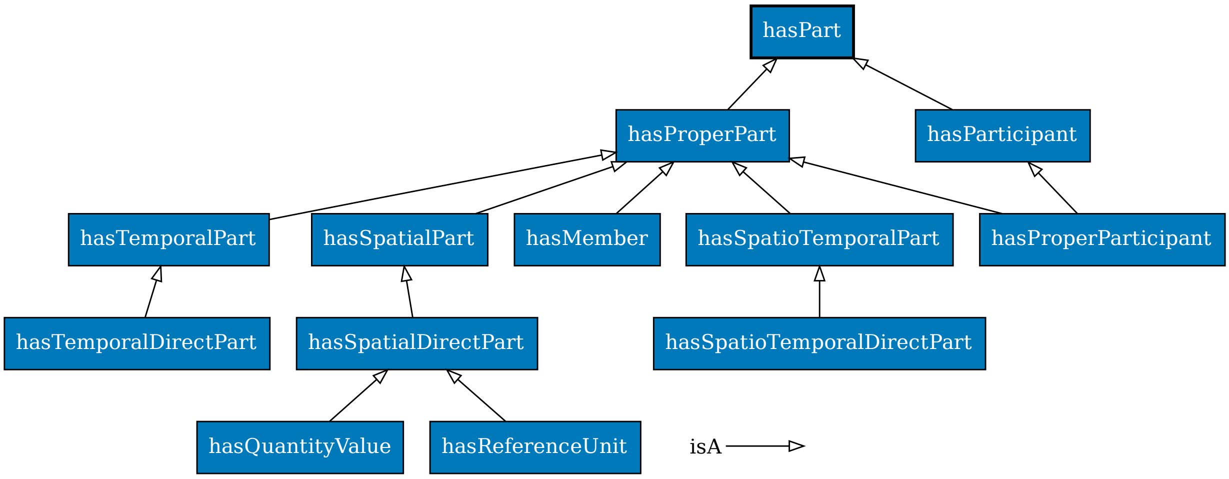


Figure 2.4: Has Part branch.

- is_a hasTemporalPart
- domain Existent
- range State

hasProperPart

IRI: http://emmo.info/emmo/top/mereotopology#EMMO_9380ab64_0363_4804_b13f_3a8a94119a76

Relations:

- is_a owl:ObjectProperty
- is_a owl:TransitiveProperty
- is_a hasPart

hasTemporalPart

IRI: http://emmo.info/emmo/top/physical#EMMO_7afbed84_7593_4a23_bd88_9d9c6b04e8f6

Elucidation: A relation that isolate a proper part that covers the total spatial extension of a whole within a time interval.

Relations:

- is_a owl:ObjectProperty
- is_a owl:TransitiveProperty
- is_a hasProperPart
- domain Item
- range Item

hasParticipant

IRI: http://emmo.info/emmo/middle/holistic#EMMO_ae2d1a96_bfa1_409a_a7d2_03d69e8a125a

Elucidation: The relation between a process and an object participating to it.

Comment: Participation is a parthood relation: you must be part (and then be connected) of the process to contribute to it.

Comment: Participation is not under direct parthood since a process is not strictly related to reductionism, but it's a way to categorize temporal regions by the interpreters.

Relations:

- is_a owl:ObjectProperty
- is_a hasPart
- domain Process
- range Participant

hasQuantityValue

IRI: http://emmo.info/emmo/middle/metrology#EMMO_8ef3cd6d_ae58_4a8d_9fc0_ad8f49015cd0

Comment: Relates a quantity to its reference unit through spatial direct parthood.

Relations:

- is_a owl:ObjectProperty
- is_a owl:FunctionalProperty
- is_a owl:InverseFunctionalProperty
- is_a owl:AsymmetricProperty
- is_a owl:IrreflexiveProperty
- is_a hasSpatialDirectPart
- domain Quantity
- range Numerical

hasSpatialPart

IRI: http://emmo.info/emmo/top/physical#EMMO_f68030be_94b8_4c61_a161_886468558054

Elucidation: A relation that isolates a proper part that extends itself in time within the overall lifetime of the whole, without covering the full spatial extension of the 4D whole (i.e. is not a temporal part).

Relations:

- is_a owl:ObjectProperty
- is_a owl:TransitiveProperty
- is_a hasProperPart
- domain Item
- range Item

hasMember

IRI: http://emmo.info/emmo/top/mereotopology#EMMO_6b7276a4_4b9d_440a_b577_0277539c0fc4

Relations:

- is_a owl:ObjectProperty
- is_a owl:AsymmetricProperty
- is_a owl:IrreflexiveProperty
- is_a hasProperPart
- domain Collection
- range Item

hasSpatioTemporalPart

IRI: http://emmo.info/emmo/top/physical#EMMO_6e046dd0_9634_4013_b2b1_9cc468087c83

Elucidation: A relation that isolates a proper part that extends itself in time through a portion of the lifetime whole.

Relations:

- is_a owl:ObjectProperty
- is_a owl:TransitiveProperty

- is_a hasProperPart
- domain Item
- range Item

hasPart

IRI: http://emmo.info/emmo/top/mereotopology#EMMO_17e27c22_37e1_468c_9dd7_95e137f73e7f

Relations:

- is_a owl:ObjectProperty
- is_a owl:TransitiveProperty
- is_a encloses
- is_a hasOverlapWith
- Inverse(mereotopology.hasOverlapWith)

hasSpatioTemporalDirectPart

IRI: http://emmo.info/emmo/middle/reductionistic#EMMO_663859e5_add3_4c9e_96fb_c99399de278d

Relations:

- is_a owl:ObjectProperty
- is_a owl:InverseFunctionalProperty
- is_a owl:AsymmetricProperty
- is_a owl:IrreflexiveProperty
- is_a hasSpatioTemporalPart

hasReferenceUnit

IRI: http://emmo.info/emmo/middle/metrology#EMMO_67fc0a36_8dcb_4ffa_9a43_31074efa3296

Comment: Relates the physical quantity to its unit through spatial direct parthood.

Relations:

- is_a owl:ObjectProperty
- is_a owl:FunctionalProperty
- is_a owl:InverseFunctionalProperty
- is_a owl:AsymmetricProperty
- is_a owl:IrreflexiveProperty
- is_a hasSpatialDirectPart
- domain Quantity
- range ReferenceUnit

hasSpatialDirectPart

IRI: http://emmo.info/emmo/middle/reductionistic#EMMO_b2282816_b7a3_44c6_b2cb_3feff1ceb7fe

Relations:

- is_a owl:ObjectProperty
- is_a owl:InverseFunctionalProperty
- is_a owl:AsymmetricProperty
- is_a owl:IrreflexiveProperty
- is_a hasSpatialPart
- domain State

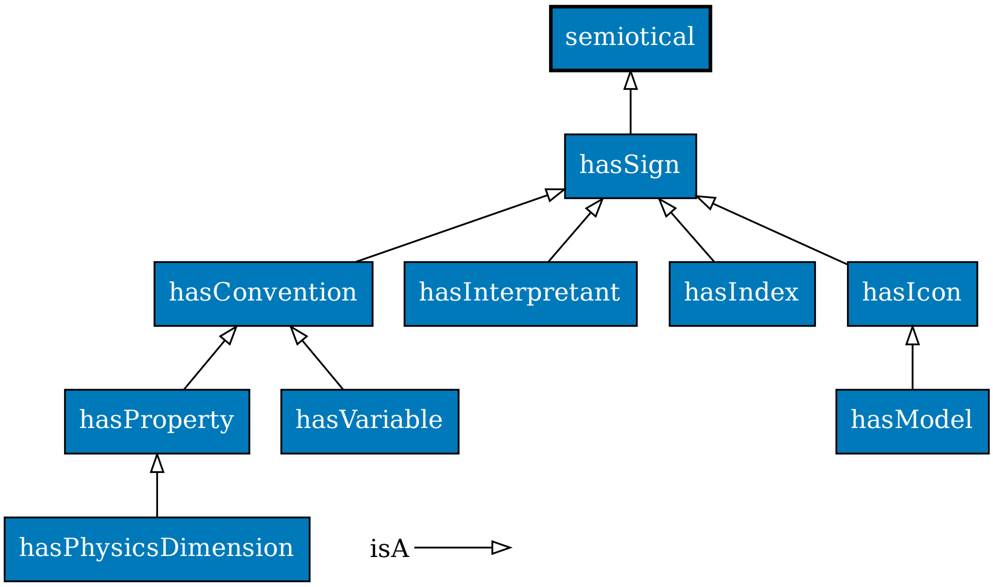


Figure 2.5: Semiotical branch.

Semiotical branch

hasPhysicsDimension

IRI: http://emmo.info/emmo/middle/metrology#EMMO_bed1d005_b04e_4a90_94cf_02bc678a8569

Relations:

- is_a owl:ObjectProperty
- is_a hasProperty
- range PhysicsDimension

hasInterpretant

IRI: http://emmo.info/emmo/top/semiotics#EMMO_7fb7fe7e_bdf9_4eeb_adad_e384dd5285c6

Relations:

- is_a owl:ObjectProperty
- is_a hasSign
- range Interpretant

hasSign

IRI: http://emmo.info/emmo/top/semiotics#EMMO_60577dea_9019_4537_ac41_80b0fb563d41

Relations:

- is_a owl:ObjectProperty
- is_a semiotical
- domain Object
- range Sign

hasIndex

IRI: http://emmo.info/emmo/top/semiotics#EMMO_297999d6_c9e4_4262_9536_bd524d1c6e21

Relations:

- is_a owl:ObjectProperty
- is_a hasSign
- range Index

hasIcon

IRI: http://emmo.info/emmo/top/semiotics#EMMO_39c3815d_8cae_4c8f_b2ff_eeba24bec455

Relations:

- is_a owl:ObjectProperty
- is_a hasSign
- range Icon

hasConvention

IRI: http://emmo.info/emmo/top/semiotics#EMMO_eb3518bf_f799_4f9e_8c3e_ce59af11453b

Relations:

- is_a owl:ObjectProperty
- is_a hasSign
- range Conventional

hasVariable

IRI: http://emmo.info/emmo/middle/math#EMMO_3446e167_c576_49d6_846c_215bb8878a55

Relations:

- is_a owl:ObjectProperty
- is_a hasConvention
- domain Mathematical
- range Variable

hasModel

IRI: http://emmo.info/emmo/middle/models#EMMO_24c71baf_6db6_48b9_86c8_8c70cf36db0c

Relations:

- is_a owl:ObjectProperty
- is_a hasIcon

hasProperty

IRI: http://emmo.info/emmo/middle/properties#EMMO_e1097637_70d2_4895_973f_2396f04fa204

Relations:

- is_a owl:ObjectProperty
- is_a hasConvention
- domain Object
- range Property

semiotical

IRI: http://emmo.info/emmo/top/semiotics#EMMO_2337e25c_3c60_43fc_a8f9_b11a3f974291

Elucidation: The generic EMMO semiotical relation.

Relations:

- is_a owl:ObjectProperty
- is_a EMMORelation
- Inverse(mereotopology.EMMORelation)

Chapter 3

EMMO Classes

emmo is a class representing the collection of all the individuals (signs) that are used in the ontology. Individuals are declared by the EMMO users when they want to apply the EMMO to represent the world.

EMMO branch

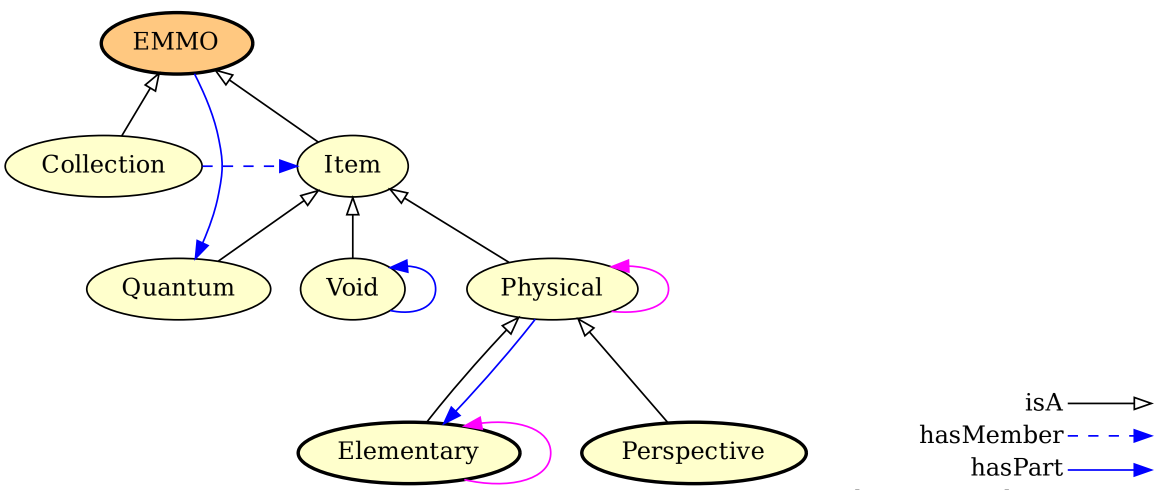


Figure 3.1: EMMO branch.

The root of all classes used to represent the world. It has two children; *collection* and *item*.

collection is the class representing the collection of all the individuals (signs) that represents a collection of non-connected real world objects.

item Is the class that collects all the individuals that are members of a set (it's the most comprehensive set individual). It is the branch of mereotopology.

Void

IRI: http://emmo.info/emmo/top/physical#EMMO_29072ec4_ffcb_42fb_bdc7_26f05a2e9873

Elucidation: A 'Item' that has no 'Physical' parts.

Comment: From Latin vacuus, "empty".

Relations:

- is_a Item
- hasPart only Void
- disjoint_with Physical, Collection

Physical

IRI: http://emmo.info/emmo/top/physical#EMMO_c5ddfdba_c074_4aa4_ad6b_1ac4942d300d

Elucidation: A 'Item' that has part some 'Elementary' and whose temporal proper parts are only 'Physical'-s (i.e. it can be perceived without interruptions in time).

Comment: A 'Physical' is the class that contains all the individuals that stand for real world objects that interact physically with the ontologist, i.e. physical objects.

A physical object must be perceived through physical interaction by the ontologist. Then the ontologist can declare an individual standing for the physical object just perceived.

Perception is a subcategory of physical interactions. It is an interaction that stimulate a representation of the physical object within the ontologist (the agent).

Comment: A 'Physical' must include at least an 'Elementary' part, and can include 'Void' parts.

A 'Physical' may include as part also the 'Void' surrounding or enclosed by its 'Physical' sub parts.

There are no particular criteria for 'Physical'-s structure, except that is made of some 'Elementary'-s as proper parts and not only 'Void'.

This is done in order to take into account the quantum nature of physical systems, in which the actual position of sub-components (e.g. electrons in an atom) is not known except for its probability distribution function (according to the Copenhagen interpretation.)

e.g. a real world object that has spatial parts an atom and a cubic light year of void, extending for some time, can be a physical object.

Comment: A 'Physical' with dimensions other than 4D cannot exist, following the restriction of the parent 'EMMO' class.

It follows from the fact that perception is always unfolding in time.

e.g. you always have an aperture time when you take a picture or measure a property. Instantaneous perceptions are idealizations (abstractions) or a very small time measurement.

Comment: From Latin physica "study of nature" (and Ancient Greek φυσικόσ, "natural").

Here the word relates to things perceived through the senses as opposed to the mind; tangible or concrete.

Comment: In the EMMO there are no relations such as occupiesSpace, since 'Physical'-s are themselves the 4D regions.

Comment: The EMMO can be used to represent real world entities as 'Physical'-s that are easy to connect to classical or quantum mechanical based models.

Classical mechanics poses no representational issues, for the EMMO: the 4D representation of 'Physical'-s is consistent with classical physics systems.

However, the representation of 'Physical'-s that are typically analized through quantum mechanics (e.g. molecules, atoms, clusters), is not straightforward.

1) De Broglie - Bohm interpretation The most simple approach is to rely on Bohmian mechanics, in which each particle is supposed to exists in a specific position between measurements (hidden variables approach), while its trajectory is calculated using a Guiding Equation based on a quantum field calculated with the Schroedinger Equation.

While this approach is really easy to implement in an ontology, since each entity has its own well defined 4D region, its mathematical representation failed to receive large consensus due to the difficulties to include relativistic effects, to be extended to subnuclear scale and the strong non-locality assumtpion of the quantum field.

Nevertheless, the Bohmian mechanics is a numerical approach that is used in electronic models to reduce the computational effort of the solution of Schroedinger Equation.

In practice, an EMMO user can declare a 'physical' individual that stand for the whole quantum system to be described, and at the same time all sub-parts individuals can be declared, having them a well defined position in time, according to De Broglie - Bohm interpretation. The Hamiltonian can be calculated by considering the sub-part individuals.

'physical'-s are then made of 'physical' parts and 'void' parts that stand for the space between 'physical'-s (e.g. the void between electrons and nucleus in an atom).

2) Copenhagen interpretation In this interpretation the properties (e.g. energy level, position, spin) of a particle are not defined in the interval between two measurements and the quantum system is entangled (i.e. properties of particles in the sysyem are correlated) and described by a global wavefunction obtained solving the Schroedinger Equation.

Upon measurement, the wavefunction collapses to a combination of close eigenstates that provide information about bservables of the system components (e.g. position, energy).

The EMMO can be used to represent 'physical'-s that can be related to Copenhagen based models. In practice, the user should follow these steps:

a) define the quantum system as a 'physical' individual (e.g. an H2 molecule) under a specific class (e.g. 'h2_molecule'). This individual is the whole.

b) define the axioms of the class that describe how many sub-parts are expected for the whole and their class types (e.g. 'h2_molecule' has axioms 'has_proper_part exactly 2 electron' and 'has_proper_part exactly 2 nucleus)

c) the user can now connect the whole to a Schroedinger equation based model whose Hamiltonian is calculated trough the information coming only from the axioms. No individuals are declared for the subparts!

d) a measurement done on the quantum system that provides information on the sub-part observables is interpreted as wavefunction collapse and leads to the end of the whole and the declaration of the sub-parts individuals which can be themselves other quantum systems

e.g. if the outer electron of the H2 molecule interacts with another entity defining its state, then the whole that stands for the entangled H2 molecule becomes a 'physical' made of an electron individual, a quantum system made of one electron and two nuclei and the void between them.

e.g. in the Born-Oppenheimer approximation the user represent the atom by un-entangling nucleus and electronic cloud. The un-entanglement comes in the form of declaration of individual as parts.

e.g. the double slit experiment can be represent in the EMMO as: a) before the slit: a 'physical' that extend in space and has parts 'electron' and 'void', called 'single_electron_wave_function'. 'electron' and 'void' are only in the axioms and not decalred individuals. b) during slit passage: a 'physical' made of one declared individual, the 'electron'. c) after the slit: again 'single_electron_wave_function' d) upon collision with the detector: 'physical' made of one declared individual, the 'electron'.

Comment: The purpose of the 'Physical' branch is to provide a representation of the real world objects, while the models used to name, explain or predict the behaviour of the real world objects lay under the 'Semiotic' branch.

More than one semiotic representation can be connected to the same 'Physical'.

e.g. Navier-Stokes or Euler equation applied to the same fluid are an example of mathematical model used to represent a physical object for some specific interpreter.

Relations:

- is_a Item
- hasPart some Elementary
- hasTemporalPart only Physical
- disjoint_with Void, Collection

Individuals:

- Universe

EMMO

IRI: http://emmo.info/emmo/top/mereotopology#EMMO_802d3e92_8770_4f98_a289_ccaaab7fdddf

Elucidation: The class representing the collection of all the individuals declared in this ontology standing for real world objects.

Comment: 'EMMO' is the disjoint union of 'Item' and 'Collection' (covering axiom).

The union implies that 'EMMO' individuals can only be 'Item' individuals (standing for self-connected real world objects) or 'Collection' individuals (standing for a collection of disconnected items).

Disjointness means that a 'Collection' individual cannot be an 'Item' individual and viceversa, representing the fact that a real world object cannot be self-connected and non-self connected at the same time.

Comment: For the EMMO ontologist the whole universe is represented as a 4D path-connected topological manifold (i.e. the spacetime).

A real world object is then a 4D topological sub-region of the universe.

A universe sub-region is isolated and defined as a real world object by the ontologist. Then, through a semiotic process that occurs at meta-ontological level (i.e. outside the ontology). an EMMO ontology entity (e.g. an OWL individual) is assigned to represent that real world object.

The fundamental distinction between real world objects, upon which the EMMO is based, is self-connectedness: a real world object can be self-connected xor not self-connected.

Comment: In the EMMO we will refer to the universe as a Minkowski space, restricting the ontology to special relativity only. However, exension to general relativity, will adding more complexity, should not change the overall approach.

Comment: Mereotopology is the fundamental logical representation used by the EMMO ontologist to characterize the universe and to provide the definitions to connect real world objects to the EMMO concepts.

Parthood relations do not change dimensionality of the real world object referred by an 'EMMO' individual, i.e. every part of a real world object always retains its 4D dimensionality.

The smallest part of a real world object (i.e. a part that has no proper parts) is referred in the EMMO by a 'Quantum' individual.

It follows that, for the EMMO, real world objects of dimensionality lower than 4D (e.g. surfaces, lines) do not exist.

Relations:

- is_a owl:Thing
- hasPart some Quantum
- equivalent_to Inverse(hasPart) value Universe
- disjoint_with
- disjoint_union_of Collection, Item

Quantum

IRI: http://emmo.info/emmo/top/mereotopology#EMMO_3f9ae00e_810c_4518_aec2_7200e424cf68

Elucidation: The class of 'EMMO' individuals that stand for real world objects that can't be further divided in time nor in space.

Example: For a physics based ontology the 'Quantum' can stand for the smallest identifiable portion of spacetime defined by the Planck limit in length (1.616e-35 m) and time (5.39e-44 s).

However, the quantum mereotopology approach is not restricted only to physics. For example, in a manpower management ontology, a 'Quantum' can stand for an hour (time) of a worker (space) activity.

Comment: A 'Quantum' is the most fundamental subclass of 'Item', since its individuals stand for the smallest possible self-connected 4D real world objects.

The quantum concept recalls the fact that there is lower epistemological limit to our knowledge of the universe, related to the uncertainity principle.

Comment: A 'Quantum' stands for a 4D real world object.

Comment: A quantum is the EMMO mereological 4D a-tomic entity.

To avoid confusion with the concept of atom coming from physics, we will use the expression quantum mereology, instead of a-tomistic mereology.

Comment: From Latin quantum (plural quanta) "as much as, so much as;", introduced in physics directly from Latin by Max Planck, 1900.

Relations:

- is_a Item
- is_a EMMO
- hasProperPart only owl:Nothing
- disjoint_with Reductionistic, Interpreter, Collection, PhysicsBasedModel, System, Semiosis, Manufacturing

Item

IRI: http://emmo.info/emmo/top/mereotopology#EMMO_eb3a768e_d53e_4be9_a23b_0714833c36de

Comment: A real world object is self-connected if any two parts that make up the whole are connected to each other (here the concept of connection is primitive).

Alternatively, using the primitive path-connectivity concept we can define a self-connected real world object as an object for which each couple of points is path-connected.

Comment: An 'Item' individual stands for a real world self-connected object which can be represented as a whole made of connected parts (e.g. a car made of components).

In the EMMO, connectivity is the topological foundation of causality.

All physical systems, i.e. systems whose behaviour is explained by physics laws, are represented only by 'Item'-s.

Members of a 'Collection' lack of causality connection, i.e. they do not constitute a physical system as a whole.

Comment: From Latin item, "likewise, just so, moreover".

Relations:

- is_a EMMO
- disjoint_with Collection
- disjoint_union_of Void, Physical

Collection

IRI: http://emmo.info/emmo/top/mereotopology#EMMO_2d2ecd97_067f_4d0e_950c_d746b7700a31

Elucidation: The class of all individuals that stand for a real world not self-connected object.

Comment: A 'Collection' individual stands for a non-self-connected real world object.

A 'Collection' individual is related to each 'Item' individuals of the collection (i.e. the members) through the membership relation.

An 'Item' individual stands for a real world self-connected object which can be represented as a whole made of connected parts (e.g. a car made of components).

Comment: Formally, 'Collection' is axiomatized as the class of individuals that hasMember some 'Item'.

A 'Collection' cannot have as member another 'Collection'.

Comment: From Latin collectio, from colligere 'gather together'.

Comment: e.g. the collection of users of a particular software, the collection of atoms that have been part of that just dissociated molecule, or even the collection of atoms that are part of a molecule considered as single individual non-connected objects and not as a mereotopological self-connected fusion.

Relations:

- is_a EMMO

- hasMember some Item
- disjoint_with Item

Elementary branch

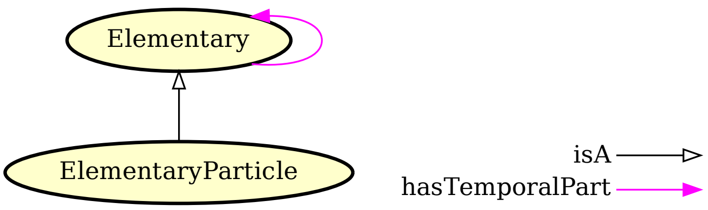


Figure 3.2: Elementary branch.

Elementary

IRI: http://emmo.info/emmo/top/physical#EMMO_0f795e3e_c602_4577_9a43_d5a231aa1360

Elucidation: The basic constituent of 'item'-s that can be proper partitioned only in time up to quantum level.

Comment: According to mereology, this should be call 'a-tomistic' in the strict etimological sense of the word (from greek, a-tomos: un-divisible).

Mereology based on such items is called atomistic mereology.

However, in order not to confuse the lexicon between mereology and physics (in which an atom is a divisible physical entity) we prefer to call it 'elementary', recalling the concept of elementary particle coming from the standard particles model.

Comment: From Latin elementārius ("elementary"), from elementum ("one of the four elements of antiquity; fundamentals")

Comment: While a 'Quantum' is a-tomistic in time and space, an 'elementary' is a-tomistic only in space, recalling the concept of elementary particle.

Relations:

- is_a Physical
- hasTemporalPart only Elementary
- hasSpatialPart only owl:Nothing
- disjoint_with Reductionistic, Interpreter, Void, Collection, PhysicsBasedModel, System

Perspective branch

Perspective

IRI: http://emmo.info/emmo/top#EMMO_49267eba_5548_4163_8f36_518d65b583f9

Elucidation: The class of individuals that stand for real world objects according to a specific representational perspective.

Comment: This class is the practical implementation of the EMMO pluralistic approach for which that only objective categorization is provide by the Universe individual and all the 'Elementary' individuals.

Between these two extremes, there are several subjective ways to categorize real world objects, each one provide under a 'Perspective' subclass.

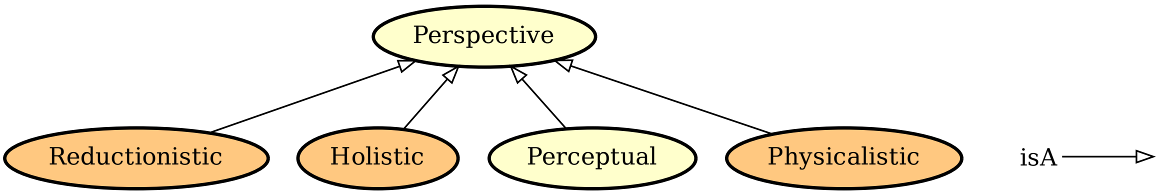


Figure 3.3: Perspective branch.

Relations:

- is_a Physical
- disjoint_with Void, Collection

Holistic branch

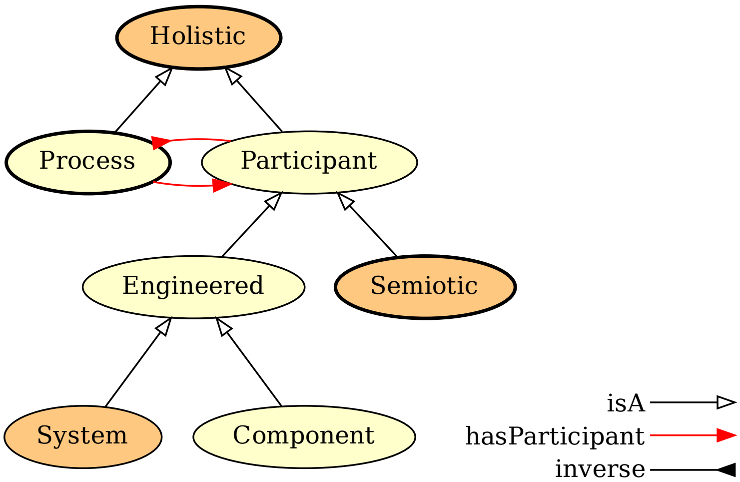


Figure 3.4: Holistic branch.

Engineered

IRI: http://emmo.info/emmo/middle/manufacturing#EMMO_86ca9b93_1183_4b65_81b8_c0fcd3bba5ad

Elucidation: A 'physical' that stands for a real world object that has been manufacturedfor a particular purpose.

Example: Car, tire, composite material.

Comment: The 'Engineered' branch represents real world objects that show some level of complexity/heterogeneity in their composition, and are made for a specific use.

Relations:

- is_a Participant
- Inverse(hasProperParticipant) some Manufacturing
- disjoint_with Void, Collection