Fundamental Categories and Relationships for intuitive querying CIDOC-CRM based repositories

Technical Report ICS-FORTH/TR-429
April 2012

Katerina Tzompanaki[1], Martin Doerr[1]

[1]Institute of Computer Science, FORTH, N. Plastira 1100, Vassilika Vouton, GR-70013 Heraklion, Greece

{katetzob@ics.forth.gr, martin@ics.forth.gr}

Abstract

The upcoming large-scale metadata repositories, semantic networks of RDF triples integrating large amounts of cultural–historical data need a rich schema upon which to build their metadata. ISO21127 (CIDOC Conceptual Reference Model) is an adequate global schema for such systems, providing the depth of classification needed to describe and integrate the information. But then raises the information access problem as querying individually hundreds of different kinds of properties leaves a huge recall gap and on the other side traditional querying techniques such as the keyword search is not adequate. On the other hand, a global restriction to "core metadata," such as Dublin Core, deprives the systems of any more advanced integration and reasoning capability. In the current document we display our proposal and design of a new query paradigm: Intuitive "fundamental" categories and relationships, as we are used to from core metadata, are presented to the user as complex deductions from a rich underlying network of more specialized actual metadata, rather than being primary documentation elements. The paths that form the Fundamental Relationships are open to verification and discussion from users and specialists of the CIDOC-CRM.

Contents

Introduction

In rich semantic networks, where the information is constructed using a schema of high complexity, useful deductions are created. In such semantic networks that provide a great level of detail a keyword based querying system or a system based on explicitly defined relationships would be insufficient. Using a specific keyword or just a flat relationship for querying would result in low recall rates, as the system would ignore all the essential information that is hidden behind deductions of relationships linking and reasoning.

Proposals to solve the problem of low recall rates in semantic networks are made towards the way of using much simpler and "poorer" schemata at the level of constructing the semantic web. In other words make the semantic network as flat as possible, such as in Dublin Core (http://dublincore.org/documents/dces/). However trying to map complicated scientific data to small schemata with poor coverage would be a failure. But even if such a project was realized such a system cannot result in good deductions and lacks the possibility of reasoning. Thus it would provide inadequate support for sophisticated queries or search precision across large datasets.

Our proposal is to simplify not the schema for constructing the semantic network, but the schema for querying the complex semantic network. To realize this, we model the world in 5 fundamental categories and we define certain generic fundamental relationships.

The 5 fundamental categories are:

- Thing (material and immaterial)
- Actor
- Place
- Time/Event
- Concept

In each category (except for the Concept), we can define an extra category to provide the Type of it, so we also have:

- Thing Type
- Actor Type
- Place Type
- Time/Event Type

This distinction is absolutely necessary because it is very common to refer to a category not by an instance of it, but by its type.

The main relationships to be modeled by the proposed fundamental categories and relationships are based on **intuition** and concern:

- Identification of real world items by real world names
- Observation and Classification of real world items
- Part-decomposition and structural properties of Conceptual & Physical Objects, Periods, Actors, Places and Times

- Participation of persistent items in temporal entities, creates a notion of history: "world-lines" meeting in space-time
- Location of periods in space-time and physical objects in space
- Influence of objects on activities and products and vice-versa
- Reference of information objects to any real-world item

Each Fundamental Category and each Fundamental Relationship is mapped to specific entities and grouping of properties of the CIDOC-CRM (http://www.cidoc-crm.org/rdfs/cidoc_crm_v5.0.2_english_label.rdfs) and the extension of it for Digital Things , the CIDOC-CRM Digital (http://www.ics.forth.gr/isl/rdfs/3D-COFORM_CRMdig.rdfs) For disputes that do not concern digital objects the part of the paths that include terms of the CIDOC-CRMdig may be omitted. In the same spirit, if other mappings are used more path-parts may be included in order to complete the needs of the certain case.

This document contains the **proposed** path for each fundamental relationship along with a description for it. For convenience we have also provided an example on how each Fundamental Relationship can be used, using real life metadata. Having in hand this document, users of the CIDOC-CRM can verify the soundness of the proposed paths and/or suggest alternatives or corrections according to their needs or their perspective. In this way we set the pedestal for a discussion to begin upon this proposed framework.

Synopsis

In this section we display a summary of all the Fundamental Relationships we use:

1. ***has type****:* denotes relations of an item to a classification, category, type, essential role or other unary property, such as a format, material, color. It generalizes over dc:type, dc:classification, dc:format, dc:language. The relationship is applicable to all FCs and has always range Concept.

2. ***is type of***: the inverse of "*has type".* The relationship is applicable to all FCs and has always domain Concept.

3. ***has part :*** the inverse of *is part of*. Denotes structural relations of an item to a narrower unit it contains. The relationship is applicable to all FCs, except for Concept. In case of Actors, one would rather speak of "*has member*", and persons are the minimal elements. Domain and range must be identical.

4. ***is part of***: denotes structural relations of an item to a wider unit it is contained in. The relationship is applicable to all FCs, except for Concept. In case of Actors, one would rather speak of "***is member of***", and persons are the minimal elements. Domain and range must be identical.

5. ***from, has generator****:* denotes the relations of an item to constituents of a context in its history which is either significant for the item, or the item is significant for the

context, "provenance" in the widest sense, including time intervals and places. In case of genealogy or group formation, natural language prefers the terms parent and founder respectively in order to refer to Actors. The relationship is a special case of has met.

6. ***is origin of, generator of***: the inverse of from, has founder or parent. In case of Actor as domain, one would rather speak of ***"is owner or creator of ".***

7. ***is similar or the same with***: denotes the symmetric relation between items that share features or are possibly identical. It is only usual for Things to document similarity manually. There exist enough comparison algorithms that deduce degrees of similarity automatically. We do not deal with these in this work.

8. ***has met****:* denotes the symmetric relation between items that were present in the same event, including time intervals and places. Applicable to any combination of FCs, except for Concepts.

9. ***refers to or is about****:* denotes the relation of an item that is information, contains information or has produced information to the item this information refers to or is about. The relation can even be extended to a Place from where such information originated.

10. ***is referred by/ is referred to at****:* the inverse of *refers to.*

11. ***borders or overlaps with***: this symmetric Relationship denotes the relationship between instances of the category place that limit with one another or overlap.

12. ***by***: denotes the active participation of an actor upon a Thing or Event

Table1 describes which of the above relationships are applicable to respective combinations of FCs as domain and range.

Domain (select)	Range(query parameter)					
	Thing	Actor	Place	Event	Time	Concept
Thing	8.has met 9.refers to or is about 10.is referred to by 3.has part 7.is similar or same with 5. from	8.has met 5.from 9.refers to or is about 10.is referred to by 12.by Used by	9.refers to 10.is referred to at 5.from Used at Created at Found or acquired at	9.refers to 10.is referred to by 5.from Destroyed in Created in Modified in Used in	5.from Destroyed on Created on Modified on Used on	1.has type

	4.is part of	Created by Modified by Found or acquired by	Was created /produced by person from Is/was located at			
Actor	8.has met 6.is owner or creator of 9. refers to 10.is referred by	4.is member of 3.has member 8. has met 5.has generator 6.is generator of 9.refers to 10.is referred by	8.has met 5.from 9.refers to 10.is referred to at	9.refers to 10.is referred to by 5.from 8.has met Was brought into existence at Was taken out of existence at Performed action at Influenced	9.refers to 5.from 8.has met Was brought into existence at Was taken out of existence at Performed action at Influenced	1.has type
Place	8.has met 6.Is origin of 9.refers to or is about 10.is referred by	8.has met 6.Is origin of 9.refers to or is about 10.is referred by 8.has met	4.is part of 3.has part 11.borders or overlaps with	9.refers to 10.is referred by 8.has met	5.from 10.refers to 8.has met	1.has type

Event	6.is origin of 10.is referred by 9.refers to or is about 8.has met created destroyed modified used	12.by 10.is referred by 9.refers to or is about 8.has met brought into existence took out of existence	9.refers to or is about 10. is referred to at 5.from	9.refers to or is about 10.is referred by 3.has part 5.from	9.refers to or is about 5.from starts ends has duration	1.has type
Time	6.is origin of	6.is origin of	6.is origin of	6.is origin of	4.is part of 3.has part	1.has type
Concept	2.is type of	2.is type of	2.is type of	2.is type of	2.is type of	1.has type 2.is type of

Table 1: Fundamental Categories – Fundamental Relationships

Notation

In this proposal we use as mentioned above two schemata, the CIDOC-CRM and the CIDOC-CRM digital. To refer to a class from the CIDOC-CRM we use the letter E (from Entity) and for the properties we use the P (from Property). Respectively we use the letter D to refer to a class from the CIDOC-CRM digital and the letter L to refer to the properties. For extentions to the already existing classes and relationships of the CIDOC-CRM we use the letter C for classes and F for properties. To elaborate this with examples:

E53.Place : a class from the CIDOC CRM

P62F.depicts: the forward link of a property of the CIDOC-CRM

P62B.is_depicted_by: the backward link of a property of the CIDOC-CRM

D1.Digital_Object :a class from the CIDOC-CRM Digital

L1F.digitized: the forward link of a property of the CIDOC-CRM Digital

L11B.was_output_of: the backward link of a property of the CIDOC-CRM Digital

A special case is the property *F1F.is_derivative_of* that hides the following rule applied in the system:

b <crmdig:L21F.used_as_derivation_source> c

b <crmdig:L22F.created_derivative> d

----

d <crm:F1F.is_derivative_of> c

This in means that if an event b used as derivation source a digital object c and it created as a derivative the digital object d, then we can deduct that the digital object d is a derivative of c.

Also we use italics for expressions such as *ObjectCreation*. At the end of the path we provide the expression that declares which classes are included or excluded in the respective expression eg

***ObjectCreation*={E63.Beginning_of_Existence NOT E67.Birth}**

In the rest of this section we display a description of the notation we use to create the paths for the FRs.

To construct a simple link that forms a triple we use ' -- ' between the subject and the predicate and ' -> ' between the predicate and the object. For example a simple triple is:

Subject -- predicate -> object

We use two types of links: direct and intermediate links. Intermediate are the links that do not connect the domain and range fundamental categories directly with one another, but through other (intermediate) entities. For presentation purposes we mark the intermediate links with blue color and the direct ones with black. We also mark with bold **blue** the intermediate classes. Classes that pose constraints (the range FC) in the path are noted with bold **red**.

For example, the semantic network might contain a direct relationship between a Thing and a Place such as:

E70.Thing-> P62F.depicts->**E53.Place**

And this means that an instance of E70 Thing refers to an instance of E53 Place.

But if the semantic network contains information about a Thing that shows features of another Thing that refers to a Place, then we may in general infer that also the first Thing depicts the Place even though not explicitly noted. So we include in our query another intermediate relation, "shows features of", among the first Thing and the Place:

E70.Thing--P130F.shows_features_of-> **E70.Thing:**

{ E70.Thing -> P62F.depicts-> **E53.Place**

}

In the example above an intermediate link is included so it is marked with blue and the intermediate class is marked with bold blue. To explain the rest of the notation consider the following example:

Subject1-- predicate1-> object1

In order to continue the path chain, and use the object1 as a subject for the next triple we add a ':' after the object1 and enclose the next triple (or block of triples) in { } brackets. So this would be:

Subject1-- predicate1-> object1:

{object1 -- predicate2 ->object2

}

To make more illustrative the fact that the next triple is one level deeper we use a tab.

Note that the next triple can continue not only with the object1 itself but also with a sub-class or a super-class of it.

We mark with green links that have as destination a **Type entity** and are used when we refer to a category through its type, eg

E70.Thing-- P62F.depicts-> **E53.Place** [--P2F.has_type -> **E55.Type**]

The use of **OR** operator can be applied either on predicates or on triples level.

When on predicates, we enclose the OR operands united with OR in brackets{ } eg

E73.Information_Object -- {P67F.refers_to OR P129F.is_about} -> **E53.Place**

When on triples again we enclose the block of triples in { } and we unite them with OR as in the following:

C1.Object -- P130F.shows_features_of -> **C1.Object:**

{C1.Object -- P31B.was_modified_by -> **E5.Event**

OR

C1.Object -- P94B.was_created_by -> **E5.Event**

}

The notation **(*relation*)**$^{[x,y]}$ implies that the relation in the parenthesis can occur minimum x times and maximum y times.

A specific case, the notation **(*relation*)**$^{[0,n]}$ means that the relation within the parenthesis may occur 0 to n times **recursively and always in the same direction**. For example **(P130F.shows_features_of)**$^{[0,n]}$ implies the path:

E70.Thing -- P130F.shows_features_of -> E70.Thing -- P130F.shows_features_of -> E70.Thing -- P130F.shows_features_of -> E70.Thing) ..

or just the E70.Thing if we have 0 occurrences of the relationship.

The **(*relation*)**$^{[x,y]}$ notation does not imply a change in the traversing direction of a path. For example **(P130F.shows_features_of)**$^{[x,y]}$ **will never** imply the path**:**

E70.Thing--P130F.shows_features_of->E70.Thing--P130B.features_are_also_found_on-> E70.Thing-- P130F.shows_features_of ->E70.Thing ..

THING

The Fundamental Category Thing is a general class that could be mapped to the CIDOC-CRM class E70.Thing. As such, this category comprises usable discrete, identifiable, instances of E77.Persistent Item that are documented as single units. However, there are some subclasses that we do not consider to be members of the FC Thing, either because they are better categorized in another FC, or because for the current discourse in the concept of the 3D-COFORM project some E70.Thing sub-classes are of no special interest. So from the general category we exclude the classes E21.Person, E55.Type, E30.Right and E41.Appelation. To formulate the result we create an expression C1.Object, to which we assign the following:

C1.Object={E70.Thing NOT E21.Person NOT E55.Type NOT E30.Right NOT E41.Appelation}.

Thing-Place

Instances of Place are geometrically identified by global coordinates or absolute reference systems. But practically one may intuitively refer to a Place by physical features that are or once have been located in a Place, such as cities or buildings. This is the reason why we also include physical features in the relationships that include the FC **Place**.

a. refers to or is about

Things often refer to or are about Places basically by their themes. So either they do so by directly referring to the Place or by being the "carrier" of the thing that refers to the Place. Copies of things that refer to or are about places also bear the same property.

Here we could make an expression that describes this combination of the interpretations of Place in order to skip the repetition of same blocks of paths :

Place= {**E26.Physical_Feature** -- P53F.has_former_or_current_location->**E53 Place**

OR

E53.Place

}

There are two properties that define the spatial containment of two Places:

- P89F.falls_within (P89B.contains)

P89F.falls_within is a super-property of *P88B.forms_part_of* . P88 also defines contextual containment relationship between the Places. Here the more general relationship P89 is enough so we use this one in the path.

In CIDOC-CRM properties that define the "refer to" or "is about" relationship are:

- P62F.depicts
- P67F.refers_to

An example from real metadata would be:

Thing** refers to or is about **Knossos

"Authentic copy of Painting of Knossos" --P130F.shows_features_of ->"**Painting of Knossos**" -- P62F.depicts -> **Knossos**

OR

"Inscription of Knossos" -- P67F.refers_to -> **Knossos**

OR

"Pottery " -- P128F.carries -> "Inscription of Knossos" -- P129F.is_about -> **Knossos**

The respective general fundamental relationship "*Thing refers to or is about Place*" is:

C1.Object --{ (P130F.shows_features_of) [0,n] OR (P130B.features_are_also_found_on) [0,n] }-> C1.Object:

{C1.Object -- { (P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n]}-> **C1.Object:**

{E24.Physical_Man-Made_Thing -- P62F.depicts **-> E53.Place**:

{E53.Place--(P89B.contains)**[0,n] -> E53.Place** [--P2F.has_type -> **E55.Type**]

}

OR

E89.Propositional_Object -- P67F.refers_to-> **E53.Place**:

{E53.Place--(P89B.contains)$^{[0,n]}$ **-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P62F.depicts **-> E26.Physical_Feature:**

{E26.Physical_Feature--P53F.has_former_or_current_location -> **E53.Place**:

{E53.Place--(P89B.contains)$^{[0,n]}$ **-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

OR

E89.Propositional_Object --P67F.refers_to -> **E26.Physical_Feature:**

{E26.Physical_Feature--P53F.has_former_or_current_location -> **E53.Place**:

{E53.Place--(P89B.contains)$^{[0,n]}$ **-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

OR

E24.Physical_Man-Made_Thing -- P128F.carries-> **E73.Information_Object:**

{ **E73.Information_Object** -- P67F.refers_to -> **E26.Physical_Feature:**

{E26.Physical_Feature--P53F.has_former_or_current_location -> **E53.Place**:

{E53.Place--(P89B.contains)$^{[0,n]}$ **-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

OR

E73.Information_Object -- P67F.refers_to -> **E53.Place**:

{E53.**Place**--(P89B.contains)$^{[0,n]}$ **-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

OR

D1.Digital_Object--(F1F.is_derivative_of)$^{[0,n]}$ **-> D1.Digital_Object:**

{D1.Digital_Object -- L11B.was_output_of -> **D7.Digital_Machine_Event:**

{D7.Digital_Machine_Event --(P9B.forms_part_of)$^{[0,n]}$ -> **D2.Digitization_Process** :

{D2.Digitization_Process -- L1F.digitized -> **E18.Physical_Thing:**

{E26.Physical_Feature-- P53F.has_former_or_current_location -> **E53.Place**:

{E53.**Place**--(P89B.contains)$^{[0,n]}$ **-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P62F.depicts -> **E53.Place**:

{E53.**Place**--(P89B.contains)$^{[0,n]}$ **-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P128F.carries -> **E73.Information_Object:**

{ **E73.Information_Object** --P67F.refers_to ->
E53.Place:

{E53.Place--(P89B.contains)$^{[0,n]}$ **-> E53.Place**
[--P2F.has_type -> **E55.Type**]

}

}

}

}

}

}

}

}

b. is referred to at

There are cases that one is interested in Things that are referred to at a certain Place. Reference implies the presence or creation of a Thing that refers to the Thing(s) in interest, as even speech is a human product, thus a Thing. So, when we talk about something, write about something or in any other way mention something the means we do it is mapped to a Thing that we create.
In this manner we connect the Thing and Place FCs not only directly but also through other Things and through Events.
In this relationship we do not use the "shows features of" property because if someone refers to a Thing, this does not mean that they also refer to any copies of it.
Moreover, we do not include here the Physical Feature notion. Mainly the reference at a Place is done with the P53F.has_former_or_current_location property of an instance of **E18.Physical_Thing** which is a super-property of the E26.Physical_Featuer. Then we also express the "is referred to at" FR through creation events that are directly connected with Places with the property P7F.took_place_at. The property P8F.took_place_on_or_within cannot be used since it connects an event with a Thing that cannot surely be placed in a Place with no respect to a time period.

In CIDOC-CRM properties that define the "is referred to" relationship are:

- P67B.is_referred_to_by
- P62B.is_depicted_by

An example from real metadata would be:

Thing** is referred to at **Crete

"Baked pottery" -- P67B.is_referred_to_by**->"Reliefs on Knossos Palace walls" --** P128B.is_carried_by **->"Knossos Palace walls" --** P53F.has_former_or_current_location -> **Knossos--** P89F.falls_within **->Crete**

OR

"Trojan horse" -- P62B.is_depicted_by->"**painting of Trojan War**"-- P53F.has_former_or_current_location -> **"Heraklion" --** P89F.falls_within **->Crete**

The respective general fundamental relationship "*Thing is referred to at Place"* is:

C1.Object -- { (P46B.forms_part_of) $^{[0,n]}$ OR (P106B.forms_part_of) $^{[0,n]}$ OR (P148B.is_component_of) $^{[0,n]}$ } -> **C1.Object:**

{C1.Object -- P67B.is_referred_to_by -> **E89.Propositional_Object:**

{E89.Propositional_Object -- P94B.was_created_by -> **E65.Creation**:

{E65.Creation --(P9B.forms_part_of)$^{[0,n]}$-> **E5.Event:**

{E65.Creation -- P7F.took_place_at ->**E53.Place:**

{E53.Place --(P89F.falls_within**)$^{[0,n]}$-> E53.Place**

[--P2F.has_type -> **E55.Type**]

}

}

}

OR

E73.Information_Object -- P128B.is_carried_by -> **E24.Physical_Man-Made_Thing:**

{E18.Physical_Thing -- P53F.has_former_or_current_location -> **E53.Place :**

{E53.Place --(P89F.falls_within**)**$^{[0,n]}$**-> E53.Place**

[--P2F.has_type -> **E55.Type**]

}

}

}

OR

C1.Object -- P62B.is_depicted_by **-> E24.Physical_Man-Made_Thing:**

{ E24.Physical_Man-Made_Thing -- P53F.has_former_or_current_location -> **E53.Place:**

{E53.Place --(P89F.falls_within**)**$^{[0,n]}$**-> E53.Place**

[--P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P108B.was_produced_by -> **E12.Production**:

{ E12.Production --(P9B.forms_part_of)$^{[0,n]}$-> **E5.Event:**

{ E5.Event -- P7F.took_place_at ->**E53.Place**:

{E53.Place --(P89F.falls_within**)**$^{[0,n]}$**-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

}

}

}

c. from

This relation returns Things that have as origin a given Place. For the Thing category, we make the general assumption that the Thing as a whole entity and its parts separately are created in the same place.

The relation, "P53F.has_former_or_current_location" is a **deduction** of "P7F.took_place_at". It is obvious that since the Event took place in a certain Place, the included objects also were once located in the same Place.

The "from (history)" relation will group the following relations:

1. The former or current location of the THING or its components.
2. The place where the THING.
3. The place where the THING was acquired or found.
4. The place associated (birth, residence) with an Actor who carried out the creation or production of the THING
5. The Place from where the THING was moved and to where the THING was moved. We can both say "Caryatid from the Parthenon" and "Caryatid from the British Museum" (as a place) and it can be differently translated as "Caryatid moved from the Parthenon" and "Caryatid moved to the British Museum".

The link (**E4 Period** --*P8 took place on or within (witnessed)* ->**E19 Physical Object**) describes that an event happens on or within a THING in which case the THING acts as place information but it does not describe any information about the origin or active participation of the THING.

The same applies for the (**E5 Event** -- *P12 occurred in the presence of (was present at)* ->**E77 Persistent Item**) link which describes the participation of a THING in an Event but it does not include any information about the origin or active participation of the THING. So these two properties cannot be included in the "from" relationship.

In CIDOC-CRM properties that define the "from" relationship are:

- P53F.has_former_or_current_location
- P7F.took_place_at
- P74F.has_current_or_former_residence
- P54F.has_current_permanent_location
- P25F.moved with P26F.moved_to and P27F.moved_from
- P24B.changed_ownership_through

An example from real metadata would be:

Thing** from **Crete

"The Dormition of the Virgin"-- P92B.was_brought_into_existence_by->**"The dormition of the Virgin Creation Event"**-- P14F.carried_out_by->**"El Greco"--** P92B.was_brought_into_existence_by **->"Birth of El Greco"--** P7F.took_place_at->**" Fodele "--** P89F.falls_within **->"Candia" --** P89F.falls_within **->Crete**

OR

"Phaistos Disc" -- P53F.has_former_or_current_location -> "**Phaistos**" -- P89F.falls_within **->Crete**

OR

"Minoan Snake Goddess faïence figurine" -- P92B.was_brought_into_existence_by-> **"Minoan Snake Goddess figurine creation event"--** P7F.took_place_at**->"Knossos"->** P89F.falls_within -> **Crete**

The respective general fundamental relationship "*Thing from Place*" is:

C1.Object -- { (P46B.forms_part_of) $^{[0,n]}$ OR (P106B.forms_part_of) $^{[0,n]}$ OR (P148B.is_component_of) $^{[0,n]}$ } -> **C1.Object:**

{C1.Object --{P53F.has_former_or_current_location OR P54F.has_current_permanent_location}-> **E53.Place**:

{E53.Place --(P89F.falls_within**)**$^{[0,n]}$**-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

OR

C1.Object -- P92B.was_brought_into_existence_by -> **E63.Beginning_of_Existence** :

{E63.Beginning_of_Existence -- (P9B.forms_part_of) $^{[0,n]}$ -> **E5.Event** :

{ E5.Event -- P7F.took_place_at -> **E53.Place**:

{E53.Place --(P89F.falls_within**)**$^{[0,n]}$**-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

OR

E7.Activity --P14F.carried_out_by **-> E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of)$^{[0,n]}$ **-> E39.Actor:**

{E39.Actor -- P74F.has_current_or_former_residence **-> E53.Place:**

{E53.Place --(P89F.falls_within**)**$^{[0,n]}$**->** **E53.Place** [--P2F.has_type -> **E55.Type**]

}

OR

E39.Actor --P92B.was_brought_into_existence_by-> **E63.Beginning_of_Existence:**

{E63.Beginning_of_Existence -- (P9B.forms_part_of) $^{[0,n]}$ -> **E5.Event:**

{ E5.Event -- P7F.took_place_at -> **E53.Place**:

{E53.Place --(P89F.falls_within**)**$^{[0,n]}$

-> E53.Place [--P2F.has_type ->

E55.Type]

}

}

}

}

}

}

OR

E19.Physical_Thing --P25B.moved_by -> **E9.Move** :

{ E9.Move -- {P26F.moved_to OR P27F.moved_from} -> **E53.Place**:

{E53.Place --(P89F.falls_within**)**$^{[0,n]}$**-> E53.Place** [--P2F.has_type ->

E55.Type]

}

}

OR

E19.Physical_Object -- P12B.was_present_at **-> C2.Finding:**

{ C2. Finding -- (P9B.forms_part_of**)**$^{[0,n]}$ **-> E5.Event:**

{ E5.Event -- P7F.took_place_at **-> E53.Place :**

{E53.Place --(P89F.falls_within)$^{[0,n]}$**-> E53.Place** [--

P2F.has_type -> **E55.Type**]

}

}

}

OR

E19.Physical_Object -- P24B.changed_ownership_through **-> E8.Acquisition:**

{ E8.Acquisition-- (P9B.forms_part_of)$^{[0,n]}$ **-> E5.Event:**

{ E5.Event -- P7F.took_place_at **->E53.Place :**

{E53.Place --(P89F.falls_within)**$^{[0,n]}$-> E53.Place** [--
P2F.has_type -> **E55.Type**]

}

}

}

}

Specializations:

In the fundamental relationship *from (history)* we can also define some sub-fundamental relationships, which contain more restricted information. This information is however commonly asked for by the users and a more specialized query may return better results as far as precision is concerned, based to what they actually want to know about. So we can define the following:

- Created in

This specialized relationship is the most important and most commonly asked, as it is mainly by the creation event of a Thing that we determine its origin. So taken an area or more generally a Place as granted, one may ask for the Things that were created, modified or produced there.

An example from real metadata would be:

Thing** created in **Greece

The Parthenon -- P92B.was_brought_into_existence_by->"**The Parthenon construction event"**-- P7F.took_place_at->**Athens** -- P89F.falls_within->**Greece**

The respective specialized fundamental relationship "*Thing created in Place*" is:

C1.Object -- { (P46B.forms_part_of) $^{[0,n]}$ OR (P106B.forms_part_of) $^{[0,n]}$ OR (P148B.is_component_of) $^{[0,n]}$ } -> **C1.Object:**

{C1.Object -- P92B.was_brought_into_existence_by -> **E63.Beginning_of_Existence*:***

{ E63.Beginning_of_Existence -- (P9B.forms_part_of) $^{[0,n)}$ -> **E5.Event**:

***{* E5.Event** -- P7F.took_place_at-> **E53.Place** :

{E53.Place --(P89F.falls_within**)$^{[0,n]}$-> E53.Place** [-- P2F.has_type -> **E55.Type**]

}

}

}

}

- Found or acquired at

Especially in the archaeological field of study, one is interested in Things found in a certain Place. This place could be a *from (history)* for this thing especially when the creation place is unknown. The same applies for the place where a Thing is acquired, which means the place where the transfer of legal ownership of a Thing takes place.

An example from real metadata would be:

Thing *found or acquired at* ***Crete***

Minoan Bee-- P12B.was_present_at->**"Minoan Bee Found Event" --** P9B.forms_part_of-> "**Excavations at Old Palace cemetery at Chrysolakkos**"-- P7F.took_place_at->"**Malia**"-- P89F.falls_within->**Crete**

OR

Modena Triptych-- P24B.changed_ownership_through ->**"Acquisition Event of Modena Triptych by the Historical Museum of Crete"** -- P7F.took_place_at ->**Heraklion --** P89F.falls_within ->**Crete**

The respective specialized fundamental relationship "*Thing found or acquired at Place"* is:

C1.Object -- { (P46B.forms_part_of) $^{[0,n]}$ OR (P106B.forms_part_of) $^{[0,n]}$ OR (P148B.is_component_of) $^{[0,n]}$ } -> **C1.Object:**

{E19.Physical_Object -- P12B.was_present_at -> **C2. Finding:**

{ C2. Finding -- (P9B.forms_part_of)$^{[0,n]}$ -> **E5.Event:**

{ E5.Event -- P7F.took_place_at -> **E53.Place :**

{E53.Place --(P89F.falls_within**)$^{[0,n]}$-> E53.Place** [-- P2F.has_type -> **E55.Type**]

}

}

}

}

OR

E19.Physical_Object -- P24B.changed_ownership_through -> **E8.Acquisition:**

{ **E8.Acquisition**-- (P9B.forms_part_of)$^{[0,n]}$ -> **E5.Event:**

{ E5.Event -- P7F.took_place_at **E53.Place :**

{E53.Place --(P89F.falls_within**)**$^{[0,n]}$**-> E53.Place** [--
P2F.has_type -> **E55.Type**]

}

}

}

}

- Was created/produced by person from

This specialized relationship gives the user the possibility to restrict the results of the "*from*" FR to only the Things that we created by an Actor from a specific Place.

An example from real metadata would be:

Thing** created/produced by person from **Crete

Alexis Zorbas, book -- P92B.was_brought_into_existence_by -> **"The writing of Alexis Zorbas" --** P14F.carried_out_by -> **"Nikos Kazantzakis" --** P92B.was_brought_into_existence_by->**"Nikos Kazantzakis Birth"** -- P7F.took_place_at **->** **"Varvaroi"**-- P89F.falls_within->**Crete**

The respective specialized fundamental relationship "*Thing created/produced by person from Place"* is:

C1.Object -- { (P46B.forms_part_of) $^{[0,n]}$ OR (P106B.forms_part_of) $^{[0,n]}$ OR (P148B.is_component_of) $^{[0,n]}$ } -> **C1.Object:**

{C1.Object -- P92B.was_brought_into_existence_by -> **E63.Beginning_of_Existence*:***

{ **E63.Beginning_of_Existence** -- (P9B.forms_part_of)[0,n] -> **E5.Event:**

{ E7.Activity -- P14F.carried_out_by -> **E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of)[0,n]->
E39.Actor:

{ E39.Actor -- P74F.has_current_or_former_residence ->
E53.Place :

{E53.Place --(P89F.falls_within**)[0,n]-> E53.Place** [--
P2F.has_type -> **E55.Type**]

}

OR

E39.Actor -- P92B.was_brought_into_existence_by ->
E63.Beginning_of_Existence :

{ E63.Beginning_of_Existence -- P7F.took_place_at ->
E53.Place:

{E53.Place --(P89F.falls_within**)[0,n]-> E53.Place** [--
P2F.has_type -> **E55.Type**]

}

}

}

}

}

- Is/was located in

One may be interested in Things that are or ever have been located in a certain Place.

An example from real metadata would be:

Thing** is/was located in **Crete

The grave of Eleutherios Venizelos -- P53F.has_former_or_current_location**->"Chania"--** P89F.falls_within->**Crete**

The respective specialized fundamental relationship "*Thing is/was located in Place*" is:

C1.Object -- { (P46B.forms_part_of) [0,n] OR (P106B.forms_part_of) [0,n] OR (P148B.is_component_of) [0,n] } -> **C1.Object:**

{E18.Physical_Thing -- P53F.has_former_or_current_location ->**E53.Place :**

{E53.Place --(P89F.falls_within**)[0,n]-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

- Moved from

With this specialization one may search for things moved from one Place.

C1.Object -- { (P46B.forms_part_of) [0,n] OR (P106B.forms_part_of) [0,n] OR (P148B.is_component_of) [0,n] } -> **C1.Object:**

{**E19.Physical_Thing** --P25B.moved_by -> **E9.Move** :

{ **E9.Move** -- P27F.moved_from -> **E53.Place**:

{**E53.Place** --(P89F.falls_within)[0,n]-> **E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

}

- Moved to

With this specialization one may search for things moved to one Place.

C1.Object -- { (P46B.forms_part_of) [0,n] OR (P106B.forms_part_of) [0,n] OR (P148B.is_component_of) [0,n] } -> **C1.Object**:

{**E19.Physical_Thing** --P25B.moved_by -> **E9.Move** :

{ **E9.Move** -- P26F.moved_to-> **E53.Place**:

{**E53.Place** --(P89F.falls_within)[0,n]-> **E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

}

Thing-Thing

a. has met

This is a general relationship meaning that a Thing has been in the same place at the same time with another Thing. This can further imply that a Thing is part of another Thing, or two Things have been present at the same Event.

An example of this really general and somehow obscure relationship can be shown in an example of metadata below:

Thing** has met **Parthenon

"pickaxe no235" -- P12B.was_present_at**->"Acropolis excavation"--**
P12F.occurred_in_the_presence_of-> **Parthenon**

The respective general fundamental relationship "*Thing has met Thing"* is:

C1.Object -- (P46B.forms_part_of)$^{[0,n]}$ -> **E18.Physical_Thing:**

{**E18.Physical_Thing** -- P12B.was_present_at -> **E5.Event :**

{ E5.Event -- (P9B.forms_part_of) $^{[0,n]}$ -> **E5.Event**:

{ E5.Event -- P12F.occurred_in_the_presence_of -> **E70.Thing:**

{E70.Thing --(P46F.is_composed_of) $^{[0,n]}$ -> **E70.Thing** [--
P2F.has_type-> **E55.Type**]

}

}

}

}

b. refers to or is about

A Thing can have as a theme or refer to another Thing, or a Thing may be created having as source a Thing that refers to another Thing. In the latter case we suppose subject preserving output events that inherit the "digitized" link. Also a Thing can be a copy of a Thing or bear similarity with a Thing that refers to another Thing and in this way they refer to the same Thing. These relationships can be modeled in the general relationship "is about".

In CIDOC-CRM properties that define the "refer to" or "is about" relationship are :

- P62F.depicts
- P67F.refers_to

An example from real metadata would be:

Thing** refers to or is about **Kazaphani Vase

"KAZAPHANI. A Middle/Late Cypriot Tomb at Kazaphani"-- P67F.refers_to-> ***Kazaphani Vase***

OR

DSC_0005.JPG --L11B.was_output_of **->"Capture photo DSC_0005 for Boat" --** P9B.forms_part_of-> **"Documentation of Laser scanning acquisition" --**P9B.forms_part_of **->"Laser scanning acquisition of Canoe-shaped vase"** -- L1F.digitized -> ***Kazaphani Vase***

OR

boat_2500png.zip--L22B.was_derivative_created_by->"**Intermediate png production for Kazaphani Vase**"--L21F.used_as_derivation_source**->"1_0.ply"**--L11B.was_output_of **->" Capture 1_0 for Boat" --** P9B.forms_part_of-> **"Detailed Sequence of shots" --** P9B.forms_part_of**->"Laser scanning acquisition of Canoe-shaped vase"** -- L1F.digitized -> ***Kazaphani Vase***

The respective general fundamental relationship "*Thing refers to or is about Thing"* is:

C1.Object -- {(P130F.shows_features_of)$^{[0,n]}$ OR (P130B.features_are_also_found_on)$^{[0,n]}$} -> **C1.Object:**

{C1.Object--{(P46F.is_composed_of) $^{[0,n]}$ OR (P106F.is_composed_of) $^{[0,n]}$ OR (P148F.has_component) $^{[0,n]}$}-> **C1.Object***:*

{ E24.Physical_Man-Made_Thing -- P62F.depicts -> **C1.Object:**

{C1.Object --{(P46F.is_composed_of) $^{[0,n]}$ OR (P106F.is_composed_of) $^{[0,n]}$ OR (P148F.has_component) $^{[0,n]}$}-> **C1.Object** [-- P2F.has_type -> **E55.Type**]

}

OR

E89.Propositional_Object --P67F.refers_to-> **C1.Object:**

{C1.Object --{(P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n]}-> **C1.Object** [-- P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P128F.carries -> **E73.Information_Object:**

{ **E73.Information_Object** --P67F.refers_to -> **C1.Object:**

{C1.Object --{(P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n]}-> **C1.Object** [-- P2F.has_type -> **E55.Type**]

}

}

OR

D1.Digital_Object--(F1F.is_derivative_of)[0,n] **-> D1.Digital_Object:**

{D1.Digital_Object -- L11B.was_output_of -> **D7.Digital_Machine_Event:**

{D7.Digital_Machine_Event --(P9B.forms_part_of)[0,n] -> **D2.Digitization_Process** :

{D2.Digitization_Process -- L1F.digitized ->**C1.Object:**

{C1.Object --{(P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n]}-> **C1.Object**[-- P2F.has_type -> **E55.Type**]

}

}

}

}

}

}

c. is referred to by

A Thing may be referred to by Things that have as theme or subject the Thing, or that refer to or are about the Thing. We may even expand the reference to Things that have as a part the Thing in reference.

In CIDOC-CRM properties that define the "is referred to by" relationship are:

- P67B.is_referred_to_by
- P62B.is_depicted_by

An example from real metadata would be:

Thing** is referred to by "**Thetis Accepting the Shield of Achilles from Vulcan" Painting

The shield of Achilles -- P67B.is_referred_to_by ->"**Thetis Accepting the Shield of Achilles from Vulcan" Painting**

The respective general fundamental relationship "*Thing is referred to by Thing"* is:

C1.Object -- { (P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n]}-> **C1.Object:**

{C1.Object--P67B.is_referred_to_by -> **E89.Propositional_Object:**

{ E89.Propositional_Object -- P148B.is_component_of->
E89.Propositional_Object[--P2F.has_type -> **E55.Type**]

OR

E73.Information_Object -- P128B.is_carried_by -> **E24.Physical_Man-Made_Thing:**

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) [0,n] ->
E24.Physical_Man-Made_Thing[--P2F.has_type -> **E55.Type**]

}

}

OR

C1.Object -- P62B.is_depicted_by-> **E24.Physical_Man-Made_Thing** :

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) [0,n] -> **E24.Physical_Man-Made_Thing**[--P2F.has_type -> **E55.Type**]

}

}

d. from

This relationship is about the origin of a Thing in regard with another Thing. This may include the primitive constituents of a Thing (*is part of* relationship), the Thing transformed into another Thing, or the Things removed from another Thing. Here we can not include the parts added to a Thing as these parts do not change the identity of the Thing. It is more suitable to include this property to the *has part* FR.

So, in this FR the Thing in the range is the wider concept of the two Things.

In CIDOC-CRM properties that define the "from" relationship are :

- P46B.forms_part_of
- P106B. forms_part_of
- P148B.is_component_of

An example from real metadata would be:

Thing** from **boat_2500png.zip

boat.mp4-- P94B.was_created_by -> **Video production for Canoe-shaped vase** -- P16F.used_specific_object -> ***boat_2500png.zip***

The respective general fundamental relationship "*Thing from Thing"* is:

C1.Object -- { (P46B.forms_part_of) [0,n] OR (P106B.forms_part_of) [0,n] OR (P148B.is_component_of) [0,n] } -> **C1.Object:**

{ **C1.Object** [--P2F.has_type -> **E55.Type**]

OR

E18.Physical_Thing -- P123B.resulted_from -> **E81.Transformation :**

{ E81.Transformation -- (P9B.forms_part_of) [0,n] -> **E81.Transformation**:

{ E81.Transformation -- P124F.transformed -> **C1.Object** ***:***

{C1.Object -- { (P46B.forms_part_of) [0,n] OR (P106B.forms_part_of) [0,n] OR (P148B.is_component_of) [0,n] } -> **C1.Object** [-- P2F.has_type -> **E55.Type**]

}

}

}

OR

E24.Physical_Man-Made_Thing -- P31B.was_modified_by-> **E11.Modification:**

{ E11.Modification-- (P9B.forms_part_of) [0,n] -> **E7.Activity**:

{

E7.Activity -- P110F.augmented-> **C1.Object** [-- P2F.has_type -> **E55.Type**]

OR

E7.Activity -- P112F.diminished-> **E18.Physical_Thing** [-- P2F.has_type -> **E55.Type**]

}

}

}

Specializations:

In the fundamental relationship *from (history)* we can also define some sub-fundamental relationships, which contain more restricted information. This information is however commonly asked for by the users and a more specialized query may return better results in terms of precision based to what they actually want to know about.

- Is part of

In this specialization the user can restrict the results of the *from* FR to only the parts that the given Thing consists of.

An example from real metadata would be:

Thing** is part of **boat_2500png.zip

Boat_photo_1.PNG -- P106B. forms_part_of -> **boat_2500.zip** **--** P106B. forms_part_of -> **boat_2500png.zip**

The respective specialized fundamental relationship "*Thing is part of Thing"* is:

C1.Object -- { (P46B.forms_part_of) [0,n] OR (P106B. forms_part_of) [0,n] OR (P148B.is_component_of) [0,n] } -> **C1.Object** [-- P2F.has_type -> **E55.Type**]

e. has part

This is a FR that indicates the part-whole relationship of a Thing with other Things. More specifically given a Thing-part we can find the Thing(s)-whole to which the Thing-part belongs. It is the reverse relationship of *Thing is part of Thing.*

An example from real metadata would be:

Thing** has part **oil jug segmentNo1

Oil jug from Aigina -- P46F.is_composed_of -> **oil jug segmentNo1**

The respective general fundamental relationship "*Thing has part Thing"* is:

C1.Object--{(P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n] }-> **C1.Object:**

{**C1.Object** [-- P2F.has_type -> **E55.Type**]

OR

E24.Physical_Man-Made_Thing -- P108B.was_augmented_by -> **E79.Part_Addition:**

{E79.Part_Addition -- (P9B.forms_part_of) [0,n] -> **E79.Part_Addition**:

{E79.Part_Addition-- P111F.added -> **E18.Physical_Thing** [-- P2F.has_type -> **E55.Type**]

}

}

}

f. is similar or same with

Similarity between two instances of the category Thing, may mean that one Thing is a copy of another, so they share the same features, or they are anchors of an annotation that indicates similarity.

An example from real metadata would be:

Thing *is similar or same with* ***Mona Lisa Painting***

Mona Lisa Authentic Copy Picture-- P130F.shows_features_of-> ***Mona Lisa Authentic Picture --*** P130F.shows_features_of-> ***Mona LisaPainting***

The respective general fundamental relationship "*Thing is similar or same with Thing*" is:

C1.Object -- { (P130F.shows_features_of) [0,n] OR (P130B.features_are_also_found_on [0,n] } -> **C1.Object:**

{ C1.Object [-- P2F.has_type -> **E55.Type**]

OR

C1.Object -- L54B.is_same-as -> **D38.Same-as :**

{ D38.Same-as -- L54F.is_same-as ->**C1.Object** [--P2F.has_type -> **E55.Type**]

}

}

Thing-Actor

a. has met

This is a general relationship meaning that a Thing has been in the same place at the same time with an Actor. This can further imply a Thing and an Actor have been present at the same Event. Here it is not right to use further inference for the Group the Actor belongs to, since it is not so probable that the Group has also been in the same event as the Actors have.

We assume that when a Thing as a whole appears at an event, then its parts also are present in the same event.

An example of this really general and somehow obscure relationship can be shown in an example of metadata below:

Thing** has met **Achilles

Paris' arrow -- P12B.was_present_at**->The Death of Achilles--**
P12F.occurred_in_the_presence_of-> **Achilles**

The respective general fundamental relationship "*Thing has met Actor*" is:

C1.Object -- (P46B.forms_part_of) $^{[0,n]}$ -> **C1.Object:**

{ **E18.Physical_Thing** -- P12B.was_present_at -> **E5.Event :**

{ E5.Event -- (P9B.forms_part_of) $^{[0,n]}$ -> **E5.Event**:

{ E5.Event -- P12F.occurred_in_the_presence_of -> **E39.Actor:**

{ E39.Actor -- (P107B.is_current_or_former_member_of)$^{[0,n]}$ ->
E39.Actor [--P2F.has_type -> **E55.Type**]

}

}

}

}

b. Is referred to by

An actor may refer to a Thing by material or immaterial Things they create and which refer to some Thing. If the Thing that is referenced is part of a bigger whole, then we assume that the reference also is valid for the bigger Thing.

In CIDOC-CRM properties that define the "is referred by" relationship are:

- P67B.is_referred_to_by
- P62B.is_depicted_by

An example from real metadata would be:

Thing** is referred to by **Salvador Dali

The ship --P129B.is_subject_of-**>"The ship painting image content"**-- P94B.was_created_by -> **"The ship painting creation event"** -- P14F.carried_out_by -> ***Salvador Dali***

OR

Clocks -- P67B.is_referred_to_by **->"The persistence of memory painting"**-- P94B.was_created_by -> **" The persistence of memory creation event"** -- P14F.carried_out_by -> ***Salvador Dali***

OR

Clock-- P62B.is_depicted_by-**>"Dance of Time II"**--P108B.was_produced_by **->"The dance of Time II sculpture production"--** P14F.carried_out_by-> ***Salvador Dali***

The respective general fundamental relationship "*Thing is referred to by Actor*" is:

C1.Object -- { (P46B.forms_part_of) [0,n] OR (P106B.forms_part_of) [0,n] OR (P148B.is_component_of) [0,n] } -> **C1.Object:**

{C1.Object-- P67B.is_referred_to_by -> **E89.Propositional_Object:**

{E89.Propositional_Object -- P94B.was_created_by -> **E65.Creation:**

{ E65.Creation -- (P9B.forms_part_of) [0,n] -> **E65.Creation**:

{ E65.Creation-- P14F.carried_out_by -> **E39.Actor:**

{ E39.Actor -- (P107B.is_current_or_former_member_of)[0,n] -> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

}

OR

E73.Information_Object -- P128B.is_carried_by -> **E24.Physical_Man-Made_Thing:**

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) 0,n) -> **E24.Physical_Man-Made_Thing**:

{ E24.Physical_Man-Made_Thing--P108B.was_produced_by -> **E12.Production:**

{ E12.Production -- (P9B.forms_part_of) [0,n] -> **E5.Event**:

{E7.Activity -- P14F.carried_out_by-> **E39.Actor:**

{ E39.Actor -- (P107B.is_current_or_former_member_of) [0,n] -> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

}

}

}

}

OR

C1.Object-- P62B.is_depicted_by-> **E24.Physical_Man-Made_Thing:**

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) $^{0,n)}$ ->
E24.Physical_Man-Made_Thing:

{ E24.Physical_Man-Made_Thing--P108B.was_produced_by ->
E12.Production:

{ E12.Production -- (P9B.forms_part_of) $^{[0,n]}$ -> **E5.Event**:

{E7.Activity -- P14F.carried_out_by-> **E39.Actor:**

{ E39.Actor --
(P107B.is_current_or_former_member_of)$^{[0,n]}$ ->
E39.Actor [--P2F.has_type -> **E55.Type**]

}

}

}

}

}

c. refers to or is about

A Thing can have as a theme or refer to an Actor, or a Thing may be created having as source a Thing that refers to an Actor. In the latter case we suppose subject preserving output events that inherit the "digitized" link, so that in the procession chain the original theme is not lost. Also a Thing can be a copy of a Thing or bear similarity with a Thing that refers to an Actor and in this way they refer to the same Actor. These relationships can be modeled in the general relationship "is about".

In CIDOC-CRM properties that define the "is about" relationship are :

- P62F.depicts
- P67F.refers_to

An example from real metadata would be:

Thing** refers to or is about **Elizabeth I

"Elizabeth I and the Three Goddesses, 1569 painting"-- P67F.refers_to-> ***Elizabeth I***

OR

eliz1.jpg --L11B.was_output_of **->"Capture photo eliz1.jpg" --** P9B.forms_part_of-> **Acquisition Event of Elizabeth I and the Three Goddesses, 1569 painting"** -- L1F.digitized -> **Elizabeth I and the Three Goddesses, 1569 painting** -- P67F.refers_to-> ***Elizabeth I***

OR

The Coronation Portrait, c. 1600, copy of 1559 --P130F.shows_features_of ->" **The Coronation Portrait, c. 1600**-- P62F.depicts-> ***Elizabeth I***

The respective general fundamental relationship "*Thing refers to or is about Actor "* is:

C1.Object --{ (P130F.shows_features_of)$^{[0,n]}$ OR (P130B.features_are_also_found_on)$^{[0,n]}$ }-> **C1.Object**:

{C1.Object--{(P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n] }-> **C1.Object:**

{E24.Physical_Man-Made_Thing -- P62F.depicts -> **E39.Actor:**

{E39.Actor -- (P107F.has_current_or_former_member) [0,n] -> **E39.Actor** [-- P2F.has_type -> **E55.Type**]

}

OR

E89.Propositional_Object --P67F.refers_to -> **E39.Actor:**

{E39.Actor -- (P107F.has_current_or_former_member) [0,n] -> **E39.Actor** [-- P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P128F.carries -> **E73.Information_Object:**

{ **E73.Information_Object** --P67F.refers_to-> **E39.Actor:**

{E39.Actor -- (P107F.has_current_or_former_member) [0,n] -> **E39.Actor** [-- P2F.has_type -> **E55.Type**]

}

}

}

}

d. by

This relation connects a Thing with the Actor who holds the property of having an active participation in the creation, modification or any other action on the Thing. We accept that parts of a Thing and the Thing itself share the same actors.

In this case we can further assume that when an actor performs an action they most probably do so as a role of being a part in a group or institution, for instance if Thomas Miller digitally acquires a photograph of a statue and Thomas Miller is a member of FORTH foundation then this photograph may also be considered as a product of FORTH.

An example from real metadata would be:

Thing** by **El Greco

"The Dormition of the Virgin"-- P92B.was_brought_into_existence_by->**"The dormition of the Virgin Creation Event"**-- P11F.had_participant **->"El Greco"**

The respective general fundamental relationship "*Thing by Actor "* is:

C1.Object--{(P106F.is_composed_of)$^{[0,n]}$ OR (P148F.has_component)$^{[0,n]}$ OR P46F.is_composed_of) $^{[0,n]}$ }-> **C1.Object :**

{ C1.Object -- {P92B.was_brought_into_existence_by OR P16B.was_used_for OR P39B.was_measured_by OR P31B.was_modified_by }-> **E7.Activity:**

{ E7.Activity -- (P9B.forms_part_of)$^{[0,n]}$ -> **E7.Activity:**

{ E7.Activity -- P14F.carried_out_by -> **E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of)$^{[0,n]}$ -> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

}

OR

E18.Physical_Thing -- P12B.was_present_at-> **C2.Finding_Event:**

{ C2.Finding_Event -- (P9B.forms_part_of)$^{[0,n]}$ -> **E5.Event:**

{ E5.Event --P11F.had_participant-> **E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of)[0,n]

-> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

}

OR

E18.Physical_Thing-- P24B.changed_ownership_through -> **E8.Acquisition:**

{ E8.Acquisition -- P22F.transferred_title_to -> **E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of)[0,n]

-> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

OR

E18.Physical_Thing--P51F.has_former_or_current_owner -> **E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of)[0,n] -> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

Specializations:

In the fundamental relationship *by* we can also define some sub-fundamental relationships, which contain more restricted information. This information is however commonly asked for by the users and a more specialized query may return better results based to what they actually want to know about.

- Used by

It is common that one is interested in the Things that were used by an Actor.

An example from real metadata would be:

Thing** used by **El Greco

PaintBrush --P16B.was_used_for-> **"View of Toledo painting event"--** P14F.carried_out_by -> **El Greco**

The respective specialized fundamental relationship "*Thing used by Actor "* is:

C1.Object --{ (P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n]}-> **C1.Object:**

{C1.Object -- P16B.was_used_for-> **E7.Activity:**

{ E7.Activity -- (P9B.forms_part_of)[0,n] -> **E5.Event:**

{E7.Activity-- P14F.carried_out_by-> **E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of)[0,n]

-> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

}

}

- Created by

The most important of the specialized relationships of the *by* FR is the "created by". With it the results are restricted to the Things an Actor has created.

An example from real metadata would be:

Thing** created by **El Greco

***View of Toledo painting*--** P94B.was_created_by -> **"View of Toledo painting event"--** P14F.carried_out_by -> **El Greco**

The respective specialized fundamental relationship "*Thing created by Actor "* is:

C1.Object --{ (P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n]}-> **C1.Object:**

{ C1.Object -- P92B.was_brought_into_existence_by->**E63.Beginning_of_Existence:**

{ E5.Event-- (P9B.forms_part_of)[0,n] -> **E5.Event:**

{E7.Activity -- P14F.carried_out_by -> **E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of)[0,n]

-> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

}

}

- Digitized by

This is a very specialized relation for objects that have undergone some digitization process and it is very common among users in this discourse.

The respective specialized fundamental relationship "*Thing digitized by Actor "* is:

C1.Object --{ (P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n]}-> **C1.Object:**

{ C1.Object -- L1B.was_digitized_by-> **D2.Digitization_Process:**

{ E5.Event-- (P9B.forms_part_of)[0,n] -> **E5.Event:**

{ E7.Activity -- P14F.carried_out_by -> **E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of)[0,n]

-> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

}

}

- Modified by

This relationship enables the user to restrict the results to the Things that have been modified by an Actor.

An example from real metadata would be:

Thing** modified by **Mustafa Kemal Atatürk

Hagia Sophia -- P31B.was_modified_by -> **"Hagia Sophia transformation to museum"--** P14F.carried_out_by -> ***Mustafa Kemal Atatürk***

The respective specialized fundamental relationship "*Thing modified by Actor "* is:

E18.Physical_Thing --(P46F.is_composed_of) [0,n]-> **E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- P31B.was_modified_by-> **E11.Modification:**

{ E5.Event-- (P9B.forms_part_of)[0,n] -> **E5.Event:**

{E7.Activity --P14F.carried_out_by-> **E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of)[0,n]

-> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

}

}

- Found or acquired by

With this relationship one can get all the Things that were found or acquired by a specific Actor.

An example from real metadata would be:

Thing** found or acquired by **Sir Arthur John Evans

clay tablet -- P12B.was_present_at -> **"excavations at Knossos"--** P14F.carried_out_by -> ***Sir Arthur John Evans***

The respective specialized fundamental relationship "*Thing found or acquired by Actor*" is:

C1.Object -- (P46F.is_composed_of) [0,n]-> **E18.Physical_Thing:**

{E18.Physical_Thing -- P12B.was_present_at->**C2.Finding_Event:**

{ C2.Finding_Event -- (P9B.forms_part_of)[0,n] -> **E5.Event:**

{E5.Event--P11F.had_participant-> **E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of)[0,n]

-> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

}

OR

E18.Physical_Thing-- P24B.changed_ownership_through -> **E8.Acquisition:**

{E5.Event-- (P9B.forms_part_of)$^{[0,n]}$ -> **E5.Event:**

{E7.Activity-- P14F.carried_out_by-> **E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of)$^{[0,n]}$

-> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

}

OR

E18.Physical_Thing--P51F.has_former_or_current_owner -> **E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of)$^{[0,n]}$ -> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

e. from

A thing can be or once have been under the possession or responsibility of one or more persons or organizations. With this relation we can track Things that have been under the possession or authority of an Actor, either individual or a group of people.

In CIDOC-CRM properties that define the “is about” relationship are :

- P49F.has_former_or_current_keeper
- P51F.has_former_or_current_owner

An example from real metadata would be:

Thing *from* **Herakleion Archeological Museum**

Snake Goddess -- P51F.has_former_or_current_owner->**Herakleion Archeological Museum**

OR

Disc of Phaestos -- P50F.has_current_keeper->**Herakleion Archeological Museum**

The respective general fundamental relationship "*Thing from Actor"* is:

C1.Object --(P46B.forms_part_of) $^{[0,n]}$ -> **E18.Physical_Thing:**

{E18.Physical_Thing--{P49F.has_former_or_current_keeper OR P51F.has_former_or_current_owner} -> **E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of)$^{[0,n]}$ -> **E39.Actor** [-- P2F.has_type -> **E55.Type**]

}

}

Thing-Event

The Thing-Event fundamental relationships can be switched to Thing-Time fundamental relationships, by further adding the CIDOC-CRM property P4F.has_time-span at the range category Event **(E5.Event** --P4F.has_time-span->**E52.Time-Span)**. This happens because Time refers to the chronological definition of Events.

a. refers to or is about

A Thing can have as a theme or refer to an Event, or a Thing may have been created having as source a Thing that refers to an Event. In the latter case we suppose subject preserving output events that inherit the "digitized" link, so that in the procession chain the original theme is not lost. Also a Thing can be a copy of a Thing or bear similarity with a Thing that

refers to an Event and in this way they refer to the same Event. These relationships can be modeled in the general relationship "is about".

In CIDOC-CRM properties that define the "is about" relationship are :

- P62F.depicts
- P67F.refers_to

An example from real metadata would be:

Thing** refers to or is about **French Revolution

"Dansons la Carmagnole song" -- P129F.is_about-> ***French Revolution***

OR

"La Liberté guidant le peuple"-- P62F.depicts-- ***French Revolution***

OR

frenchRevolPhotos.zip--L22B.was_derivative_created_by->"**Archiving revolution photos**"--L21F.used_as_derivation_source**->"revolution_photo.png"**--L11B.was_output_of **->" Capture 1_0 for Revolution painting" --** (P9B.forms_part_of)[0,n]-> **"Detailed Sequence of shots" --(**P9B.forms_part_of)[0,n] **->"Laser scanning acquisition of La Liberté guidant le peuple"** -- L1F.digitized -> ***French Revolution***

The respective general fundamental relationship "*Thing refers to or is about Event*" is:

C1.Object -- { (P130F.shows_features_of) [0,n] OR (P130B.features_are_also_found_on) [0,n] } -> **C1.Object:**

{C1.Object --{ (P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n]}-> **C1.Object:**

{ E24.Physical_Man-Made_Thing -- P62F.depicts -> **E5.Event:**

{E5.Event --(P9B.forms_part_of)[0,n] -> **E5.Event** [-- P2F.has_type -> **E55.Type**]

}

OR

E89.Propositional_Object --P67F.refers_to -> **E5.Event:**

{E5.Event --(P9B.forms_part_of)$^{[0,n]}$ -> **E5.Event** [-- P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P128F.carries -> **E73.Information_Object:**

{ **E73.Information_Object** --P67F.refers_to -> **E5.Event:**

{E5.Event --(P9B.forms_part_of)$^{[0,n]}$ -> **E5.Event** [-- P2F.has_type -> **E55.Type**]

}

}

OR

D1.Digital_Object--(F1F.is_derivative_of)$^{[0,n]}$ **-> D1.Digital_Object:**

{D1.Digital_Object -- L11B.was_output_of -> **D7.Digital_Machine_Event:**

{D7.Digital_Machine_Event --(P9B.forms_part_of)$^{[0,n]}$ -> **D2.Digitization_Process** :

{D2.Digitization_Process -- L1F.digitized -> **E18.Physical_Thing:**

{ E24.Physical_Man-Made_Thing -- P62F.depicts -> **E5.Event:**

{E5.Event --(P9B.forms_part_of)$^{[0,n]}$ -> **E5.Event** [-- P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P128F.carries ->

E73.Information_Object:

{ **E73.Information_Object** --P67F.refers_to ->

E5.Event:

{E5.Event --(P9B.forms_part_of)[0,n] ->

E5.Event [-- P2F.has_type -> **E55.Type**]

}

}

}

}

}

}

}

}

b. is referred to at

Things may be referred by Events during which Things that refer to them were created.

In CIDOC-CRM properties that define the "is referred to at" relationship are:

- P67B.is_referred_to_by
- P62B.is_depicted_by

An example from real metadata would be:

Thing** is referred to at **The Argo painting Event

The Argo ship -- P62B.is_depicted_by -> **"The Argo" painting --** P108B.was_produced_by-> **The Argo painting Event**

The respective general fundamental relationship "*Thing is referred to at Event"* is:

C1.Object -- { (P46B.forms_part_of) [0,n] OR (P106B.forms_part_of) [0,n] OR (P148B.is_component_of) [0,n] } -> **C1.Object:**

{C1.Object--P67B.is_referred_to_by -> **E89.Propositional_Object:**

{E89.Propositional_Object -- P94B.was_created_by -> **E65.Creation:**

{ E65.Creation -- (P9B.forms_part_of) [0,n] -> **E5.Event** [--P2F.has_type -> **E55.Type**]

}

OR

E73.Information_Object -- P128B.is_carried_by -> **E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) 0,n) -> **E24.Physical_Man-Made_Thing**:

{ E24.Physical_Man-Made_Thing--P108B.was_produced_by -> **E12.Production:**

{ E12.Production -- (P9B.forms_part_of) [0,n] -> **E5.Event** [--P2F.has_type -> **E55.Type**]

}

}

}

OR

C1.Object-- P62B.is_depicted_by-> **E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing--P108B.was_produced_by -> **E12.Production:**

{ E12.Production -- (P9B.forms_part_of) [0,n] -> **E5.Event** [--P2F.has_type -> **E55.Type**]

}

}

}

c. Has met

Things can be linked with certain events despite the fact that the kind of link may not be known to the user. So questions concerning the placement of Things in events may be as generic as a generic *from.* More specific constraints on this relationship can be applied at the specializations of the *from* relationship.

An example from real metadata would be:

Thing** from **French Revolution

Charleville-- P12B.was_present_at-> ***French Revolution***

The respective general fundamental relationship "*Thing from Event*" is:

C1.Object -- { (P46B.forms_part_of) [0,n] OR (P106B.forms_part_of) [0,n] OR (P148B.is_component_of) [0,n] } -> **C1.Object:**

{ C1.Object -- P12B.was_present_at-> **E5.Event:**

{E5.Event--(P9B.forms_part_of)[0,n]->**E5.Event** [--P2F.has_type -> **E55.Type**]

}

}

Specializations:

In the fundamental relationship *from* we can also define some sub-fundamental relationships, which contain more restricted information. This information is however commonly asked for by the users and a more specialized query may return better results based to what they actually want to know about.

- Destroyed in

In this specialization we are interested in the Things that were destroyed in an Event. We suppose that the destruction of a thing further means the destruction of its parts.

An example from real metadata would be:

Thing** destroyed in **Rome earthquake in 1349

outer south side of Colosseum -- P46B.forms_part_of -> **Colosseum --** P13B.was_destroyed_by-> **Rome earthquake in 1349**

The respective specialized fundamental relationship "*Thing destroyed in Event*" is:

C1.Object -- { (P46B.forms_part_of) $^{[0,n]}$ OR (P106B. forms_part_of) $^{[0,n]}$ OR (P148B.is_component_of) $^{[0,n]}$ } -> **C1.Object:**

{C1.Object-- P93B.was_taken_out_of_existence_by-> **E64.End_of_Existence*:***

{E64.End_of_Existence --(P9B.forms_part_of)$^{[0,n]}$->

E5.Event [--P2F.has_type -> **E55.Type**]

}

}

- Created in

Things created in Events are of special interest for researchers and scientists so this specialization is very useful. We take as granted that the parts of a Thing are created during the creation of the Thing.

An example from real metadata would be:

Thing** created in **Laser scanning acquisition of Canoe-shaped vase

DSC_0005.JPG -- P92B.was_brought_into_existence_by **->"Capture photo DSC_0005 for Boat" --** P9B.forms_part_of-> **"Documentation of Laser scanning acquisition" --** P9B.forms_part_of **->"Laser scanning acquisition of Canoe-shaped vase"**

The respective specialized fundamental relationship "*Thing created in Event"* is:

C1.Object -- { (P46B.forms_part_of) $^{[0,n]}$ OR (P106B. forms_part_of) $^{[0,n]}$ OR (P148B.is_component_of) $^{[0,n]}$ } -> **C1.Object:**

{C1.Object -- P92B.was_brought_into_existence_by-> **E63.Beginning_of_Existence*:***

***{* E63.Beginning_of_Existence** --(P9B.forms_part_of)$^{[0,n]}$->**E5.Event**[-- P2F.has_type-> **E55.Type**]

}

}

- Modified in

In this specialization we are interested in Things that have been modified during an event.

An example from real metadata would be:

Thing** modified in **Rome Palace Construction

tumbled stone from Colosseum -- P31B.was_modified_by -> **Colosseum Stone reusage--** P9B.forms_part_of -> **Rome Palace Construction**

The respective specialized fundamental relationship "*Thing modified in Event*" is:

C1.Object -- (P46B.forms_part_of) $^{[0,n]}$ -> **C1.Object:**

{E24.Physical_Man-Made_Thing--P31B.was_modified_by-> **E11.Modification:**

{**E11.Modification**--(P9B.forms_part_of)$^{[0,n]}$->

E5.Event [--P2F.has_type -> **E55.Type**]

}

- Used in

Often one is interested in the Things that were used during an Event.

An example from real metadata would be:

Thing** used in **Greek Revolution of 1821

Greek rifle-- P16B.was_used_for->**"Battle of Peta"--** P9B.forms_part_of->**"Greek Revolution of 1821"**

The respective specialized fundamental relationship "*Thing used in Event*" is:

C1.Object -- {(P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n]}-> **C1.Object:**

{**C1.Object** -- P16B.was_used_for->**E7.Activity:**

{**E7.Activity**--(P9B.forms_part_of)[0,n]->**E5.Event** [--P2F.has_type -> **E55.Type**]

}

}

- Digitized in

This specialization is especially useful for users in the concept of digitization processes. With this can one retrieve the events that were responsible for the digitization of the specific Thing.

The respective specialized fundamental relationship "*Thing digitized in Event*" is:

C1.Object --(P46B.forms_part_of)[0,n] -> **C1.Object:**

{**C1.Object** -- L1B.was_digitized_by-> **D2.Digitization_Process:**

{ **D2.Digitization_Process** --(P9B.forms_part_of)[0,n]-> **E5.Event** [--P2F.has_type -> **E55.Type**]

}

}

Thing-Concept

a. has type

This is a simple relationship that expresses the Concept a.c.a. Type of the Thing. This relationship may be of great importance when we want to gather all the Things that have the same type and then use them in conjunction with another FC regarding Things.

An example from real metadata would be:

Thing** has type **weapon

Charleville -- P2F.has_type-> **weapon**

The respective general fundamental relationship "*Thing has type Concept*" is:

C1.Object -- {(P46F.is_composed_of)$^{[0,n]}$ OR (P106F.is_composed_of)$^{[0,n]}$ OR (P148F.has_component)$^{[0,n]}$**}-> C1.Object:**

{ C1.Object -- P2F.has_type**-> E55.Type:**

{E55.Type --{(P127 has broader term)$^{[0,n]}$ OR (P2F.has_type)$^{[0,n]}$}-> **E55.Type**

}

OR

C1.Object --P45F. consists_of**-> E57.Material:**

{ E57.Material --{(P127 has broader term)$^{[0,n]}$ OR (P2F.has_type)$^{[0,n]}$}-> **E55.Type**

}

OR

C1.Object -- P92B.was_brought_into_existence_by**-> E7.Activity:**

{ E7.Activity --(P9B.forms_part_of)$^{[0,n]->}$**E7.Activity:**

{E7.Activity--P33F.used_specific_technique**->** **E29.Design_or_Procedure:**

{E29.Design_or_Procedure-- P68F.foresees_use_of-> E57.Material:

{ E57.Material --{(P127 has broader term)$^{[0,n]}$ OR (P2F.has_type)$^{[0,n]}$}-> **E55.Type**

}

}

OR

E11.Modification -- P126F.employed -> E57.Material:

{ E57.Material --{(P127 has broader term)$^{[0,n]}$ OR (P2F.has_type)$^{[0,n]}$}-> **E55.Type**

}

}

}

}

Specialization:

- Is made of

It is typical that a user may be interested in objects that are not generally of some type, but that are made of some specific material. For this case, they can use the specialization of the *has type* FR, **Thing is made of Concept.**

C1.Object -- {(P46F.is_composed_of)[0,n] OR (P106F.is_composed_of)[0,n] OR (P148F.has_component)[0,n]**}-> C1.Object:**

{ C1.Object --P45F. consists_of-> E57.Material

OR

C1.Object -- P92B.was_brought_into_existence_by**-> E7.Activity:**

{E7.Activity --(P9B.forms_part_of)[0,n]->**E7.Activity:**

{E7.Activity--P33F.used_specific_technique**->**
E29.Design_or_Procedure:

{E29.Design_or_Procedure-- P68F.foresees_use_of->
E57.Material

}

OR

E11.Modification -- P126F.employed -> E57.Material

}

}

}

PLACE

The fundamental category place comprises geometric extents in space, on earth and on objects, which are independent of matter or temporal changes. Nevertheless intuitively Physical Features that are located in a Place are themselves considered to be Places. Such examples can be cities or settlements. Thus, by referring to Knossos we don't only speak of the ancient city but also the place where the city was located. In relation with the other FCs Place most commonly is used to identify where events take place or where material items, living or lifeless have been once or are permanently located.

Place-Place

a. has part

Given a Place one may be interested in broader or narrower Places it has some connection with. In this relationship, we are interested in the broader Places, in other words the Places that the Place forms part of.

In CIDOC-CRM properties that define the "has part" relationship are:

- P89B.contains

An example from real metadata would be:

Place** has part **Crete

Europe-- P88F.consists_of -> **Greece** -- P88F.consists_of -> **Crete**

The respective general fundamental relationship "*Place has part Place"* is:

E53.Place--(P89B.contains)$^{[0,n]}$ **-> E53.Place** [--P2F.has_type -> **E55.Type**]

b. Is part of

This is the backward relationship of the "has part" relationship. Thus, here we are interested in Places that the given Place consists of as a part-whole relationship.

In CIDOC-CRM properties that define the "is part of" relationship are :

- P89F.falls_within

An example from real metadata would be:

Place** is part of **Europe

Sicily -- P88B.forms_part_of->**Italy --** P88B.forms_part_of->**Europe**

The respective general fundamental relationship "*Place is part of or limit of Place"* is:

E53.Place --(P89F.falls_within)$^{[0,n]}$ -> **E53.Place** [--P2F.has_type -> **E55.Type**]

c. Borders or overlaps with

To complete the relationships that a Place can have with another Place, there is also the proximity property among Places. So, with this relationship the user can find out which Places the given Place borders or overlaps with.

The following example shows how useful is for this relationship to also use the whole-part transitivity. Of course in the end we run the risk of concluding to that a Place borders or overlaps with itself or a broader Place, which actually is the fundamental Relationship Place *has part* Place.

An example from real metadata would be:

Place** borders or overlaps with **Asia

Greece -- P89B.contains ->**Thrace --** P122F.borders_with -> **Turkey --** P89F.falls_within ->**Asia**

OR

Asia -- P89B.contains ->**Syria --** P122F.borders_with -> **Turkey --** P89F.falls_within ->**Asia**

The respective general fundamental relationship "*Place borders with Place"* is:

E53.Place --(P89B.contains)$^{[0,n]}$ -> **E53.Place** *:*

{**E53.Place --** {P122F.borders_with **OR** P121F.overlaps_with **}-> E53.Place:**

{E53.Place --(P89F.falls_within)$^{[0,n]}$ -> **E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

Place-Thing

a. refers to

A place can refer to a Thing by artefacts created or located in that place that have as subject or depict the thing. Also a Place refers to a Thing when the Actor that owns or keeps the Thing has or has ever had their residence in that Place. So with this relationship we can find out about the Places where a Thing is referred to at.

In CIDOC-CRM properties that define the "refer to" or "is about" relationship are:

- P62F.depicts
- P67F.refers_to

An example from real metadata would be:

Place *refers to* ***The Cross of Jesus Christ***

France -- P88F.consists_of->**Colmar--**P74B.is_current_or_former_residence_of -> **Musee d'Unterlinden --** P51B.is_former_or_current_owner_of -> **the crusifixion of christ painting** P67F.refers_to -> **The Cross of Jesus Christ**

The respective general fundamental relationship "*Place refers to Thing"* is:

E53.Place--(P89B.contains) [0,n] -> **E53.Place:**

{E53.Place-- P53B.is_former_or_current_location_of -> **E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- P62F.depicts -> **C1.Object:**

{C1.Object --{(P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n]}-> **C1.Object** [-- P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing --P128F.carries-> **E73.Information_Object:**

{E73.Information_Object -- P67F.refers_to -> **C1.Object:**

{C1.Object --{(P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n]}-> **C1.Object** [-- P2F.has_type -> **E55.Type**]

}

}

}

OR

E53.Place-- P74B.is_current_or_former_residence_of -> **E39.Actor:**

{E39.Actor-- {P49B.is_former_or_current_keeper_of OR P51B.is_former_or_current_owner_of **} -> E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- P62F.depicts -> **C1.Object:**

{C1.Object --{(P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n]}-> **C1.Object** [-- P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing --P128F.carries-> **E73.Information_Object:**

{E73.Information_Object -- P67F.refers_to ->**C1.Object:**

{C1.Object --{(P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n] } -> **C1.Object** [-- P2F.has_type -> **E55.Type**]

}

}

}

}

OR

E53.Place-- P7B.witnessed-> **E5.Event:**

{ E5.Event --(P9F.consists_of)[0,n]-> **E5.Event:**

{ E65.Creation -- P94F.has_created**-> E18.Physical_Thing:**

{ E89.Propositional_Object --P67F.refers_to -> **C1.Object:**

{C1.Object --{(P46F.is_composed_of) [0,n] OR
(P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n]}->
C1.Object [-- P2F.has_type -> **E55.Type**]

}

}

OR

E12.Production -- P108F.has_produced **-> E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- P62F.depicts -> **C1.Object:**

{C1.Object --{(P46F.is_composed_of) [0,n] OR
(P106F.is_composed_of) [0,n] OR (P148F.has_component) [0,n]}->
C1.Object [-- P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing --P128F.carries->
E73.Information_Object:

{E73.Information_Object -- P67F.refers_to -> **C1.Object:**

{C1.Object --{(P46F.is_composed_of) [0,n] OR
(P106F.is_composed_of) [0,n] OR
(P148F.has_component) [0,n]}-> **C1.Object** [--
P2F.has_type -> **E55.Type**]

}

}

}

}

b. is referred to by

A Place may be referred to by Things that have as theme or subject the Place, or that refer to or are about the Place. We may even expand the reference to Places that contain the Place, both geographically and by context (so here we use the P88 CIDOC-CRM property).

In CIDOC-CRM properties that define the "is referred by" relationship are:

- P67B.is_referred_to_by
- P62B.is_depicted_by

An example from real metadata would be:

Place** is referred to by **Guernica painting

Guernica-- P67B.is_referred_to_by -> **"Guernica painting theme"** -- P128B.is_carried_by -> **Guernica Painting**

The respective general fundamental relationship "*Place is referred to by Thing"* is:

E53.Place -- (P89F.falls_within)$^{[0,n]}$**-> E53.Place:**

{E53.Place --P67B.is_referred_to_by -> **E89.Propositional_Object:**

{ E89.Propositional_Object --(P148B.is_component_of) $^{[0,n]}$->
E89.Propositional_Object[--P2F.has_type -> **E55.Type**]

OR

E73.Information_Object -- P128B.is_carried_by -> **E24.Physical_Man-Made_Thing:**

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of)$^{[0,n]}$ ->
E24.Physical_Man-Made_Thing[--P2F.has_type -> **E55.Type**]

}

}

OR

E53.Place -- P62B.is_depicted_by-> **E24.Physical_Man-Made_Thing** :

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of)$^{[0,n]}$ ->
E24.Physical_Man-Made_Thing[--P2F.has_type -> **E55.Type**]

}

}

c. has met

This relationship is used to connect a place with things that have once been located in a places or have been created there so consequently have been once located there. Movement of a Thing from one Place to another also implies that the Thing has once been at both Places. Moreover we can make the assumption that a Thing has been located in the same place as their keeper or owner.

An example from real metadata would be:

***Place** has met Caryatid*

Athens -- P53B.is_former_or_current_location_of -> **Caryatid**

OR

London -- P74B.is_current_or_former_residence_of ->**British Museum**--
P49B.is_former_or_current_keeper_of -> **Caryatid**

The respective general fundamental relationship "*Place has met Thing"* is:

E53.Place--(P89B.contains) $^{[0,n]}$ -> **E53.Place:**

{E53.Place-- P53B.is_former_or_current_location_of -> **E24.Physical_Man-Made_Thing** :

{ E24.Physical_Man-Made_Thing -- (P46F.is_composed_of) $^{[0,n]}$ ->
E24.Physical_Man-Made_Thing[--P2F.has_type -> **E55.Type**]

}

OR

E53.Place --P7B.witnessed->**E5.Event:**

{E5.Event --(P9F.consists_of)$^{[0,n]}$-> **E5.Event:**

{E5.Event--P98F.brought_into_life-> **C1.Object:**

{ C1.Object --{(P46B.forms_part_of)$^{[0,n]}$ OR (P106B.forms_part_of) $^{[0,n]}$ OR (P148B.is_component_of) $^{[0,n]}$}-> **C1.Object** [--P2F.has_type -> **E55.Type**]

}

OR

E11.Modification-- P31F.has_modified -> **C1.Object:**

{ C1.Object --{(P46B.forms_part_of)$^{[0,n]}$ OR (P106B.forms_part_of) $^{[0,n]}$ OR (P148B.is_component_of) $^{[0,n]}$}-> **C1.Object** [--P2F.has_type -> **E55.Type**]

}

OR

E9.Move -- P25F.moved **->** **E19.Physical_Object:**

{ E19.Physical_Object --(P46B.forms_part_of)$^{[0,n]}$ ->
E19.Physical_Object[--P2F.has_type -> **E55.Type]**

}

}

}

OR

E53.Place-- P74B.is_current_or_former_residence_of -> **E39.Actor:**

{E39.Actor-- {P49B.is_former_or_current_keeper_of OR

P51B.is_former_or_current_owner_of**} -> E18.Physical_Thing:**

{E18.Physical_Thing --(P46B.forms_part_of)[0,n] ->

E18.Physical_Thing [--P2F.has_type -> **E55.Type]**

}

}

}

Specializations:

The user frequently may only be interested in things that are products of a place so a specialization of the former relationship is:

- is_origin_of

An example from real metadata would be:

***Place** is origin of* **Great Pyramid of Giza**

Egypt -- P7B.witnessed->**The building of the Great Pyramid of Giza** --
P98F.brought_into_life-> **Great Pyramid of Giza**

OR

Fodele-- P7B.witnessed->**Birth of El Greco** --P98F.brought_into_life-> **Great Pyramid of Giza**

The respective specialized fundamental relationship "*Place is origin of Thing"* is:

E53.Place--(P89B.contains) [0,n] -> **E53.Place:**

{E53.Place --P7B.witnessed->**E5.Event:**

{E5.Event --(P9F.consists_of)[0,n]-> **E5.Event:**

{E5.Event--P98F.brought_into_life-> **C1.Object:**

{ C1.Object --{(P46B.forms_part_of)[0,n] OR (P106B.forms_part_of)[0,n] OR (P148B.is_component_of) [0,n]}-> **C1.Object** [--P2F.has_type -> **E55.Type**]

}

}

}

}

Place-Actor

a. refers to

A place can refer to an actor by artefacts created or located in that place that have as subject or depict the actor. So with this relationship we can find out about the Places where an Actor is referred to at.

In CIDOC-CRM properties that define the “refer to” or “is about” relationship are :

- P62F.depicts
- P67F.refers_to

An example from real metadata would be:

***Place** refers to* **El Greco**

Metropolitan Museum of Art -- P53B.is_former_or_current_location_of -> **El Greco portrait** -- P62F.depicts -> **El Greco**

OR

Spain -- P88F.consists_of->**Toledo**-- P7B.witnessed-> **El Greco portrait Creation Event** -- P108F.has_produced -> **El Greco portrait--** P62F.depicts -> **El Greco**

The respective general fundamental relationship "*Place refers to Actor*" is:

E53.Place--(P89B.contains)$^{[0,n]}$ -> **E53.Place:**

{E53.Place-- P53B.is_former_or_current_location_of -> **E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- P62F.depicts ->**E39.Actor:**

{E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$->

E39.Actor [--P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing --P128F.carries->
E73.Information_Object:

{E73.Information_Object -- P67F.refers_to->**E39.Actor:**

{E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$->

E39.Actor [--P2F.has_type -> **E55.Type**]

}

}

}

OR

E53.Place-- P7B.witnessed-> **E5.Event:**

{ E5.Event --(P9F.consists_of)$^{[0,n]}$-> **E5.Event:**

{E65.Creation -- P94F.has_created**-> E18.Physical_Thing:**

{ E89.Propositional_Object --P67F.refers_to -> **E39.Actor:**

{E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$->

E39.Actor [--P2F.has_type -> **E55.Type**]

}

}

OR

E12.Production -- P108F.has_produced **-> E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- P62F.depicts ->**E39.Actor:**

{E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$->

E39.Actor [--P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing --P128F.carries->
E73.Information_Object:

{E73.Information_Object -- P67F.refers_to ->**E39.Actor:**

{E39.Actor --
(P107F.has_current_or_former_member)$^{[0,n]}$->

E39.Actor [--P2F.has_type -> **E55.Type**]

}

}

}

}

}

}

b. is referred to by

A Place may be referred to by Actors though artifacts that they create and have as a theme or subject or refer to the Place.

In CIDOC-CRM properties that define the "is referred to by" relationship are:

- P67B.is_referred_to_by
- P62B.is_depicted_by

An example from real metadata would be:

Place** is referred to by **Kazantzakis

Russia -- P129B.is_subject_of ->**"Travelling: Russia"**-- P94B.was_created_by -> "**Travelling: Russia writing**" -- P14F.carried_out_by-->**Kazantzakis**

The respective general fundamental relationship "*Place is referred to by Actor*" is:

E53.Place-- (P89F.falls_within)$^{[0,n]}$ -> **E53.Place:**

{E53.Place --P67B.is_referred_to_by-> **E89.Propositional_Object:**

{E89.Propositional_Object -- P94B.was_created_by ->**E65.Creation**:

{E65.Creation -- (P9B.forms_part_of)$^{[0,n]}$-> **E7.Activity:**

{E7.Activity -- P14F.carried_out_by->**E39.Actor:**

{E39.Actor --
(P107F.has_current_or_former_member)[0,n]->
E39.Actor [--P2F.has_type -> **E55.Type**]
}

}

}

OR

E73.Information_Object -- P128B.is_carried_by -> **E24.Physical_Man-Made_Thing:**

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) 0,n) ->
E24.Physical_Man-Made_Thing:

{E24.Physical_Man-Made_Thing --
P108B.was_produced_by **-> E12.Production:**

{ E12.Production -- (P9B.forms_part_of)[0,n]->
E5.Event:

{E7.Activity -- P14F.carried_out_by->
E39.Actor:

{E39.Actor --
(P107F.has_current_or_former_member)[0,n]->
E39.Actor [--P2F.has_type ->
E55.Type]
}

}

}

}

}

}

OR

E53.Place -- P62B.is_depicted_by-> **E24.Physical_Man-Made_Thing**:

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) $^{0,n)}$ **->**
E24.Physical_Man-Made_Thing:

{E24.Physical_Man-Made_Thing -- P108B.was_produced _by **->**
E12.Production:

{E12.Production -- (P9B.forms_part_of)$^{[0,n]}$-> **E5.Event:**

{E7.Activity -- P14F.carried_out_by->**E39.Actor:**

{E39.Actor --
(P107F.has_current_or_former_member)$^{[0,n]}$
->**E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

}

}

}

c. has met

This relationship is used to connect a place with persons or groups of people that are born (formed) in a place or have had the place as residence. Also when an event that is witnessed by an Actor takes place at a certain Place, then we deduct that also the Actor has met the Place.

In CIDOC-CRM properties that define the "has met" relationship are :

- P74B.is_current_or_former_residence_of
- P7B.witnessed

An example from real metadata would be:

***Place** has met* **El Greco**

Crete -- P88F.consists_of->**Fodele--** P7B.witnessed->**Birth of El Greco --** P98F.brought_into_life-> **El Greco**

OR

Toledo -- P74B.is_current_or_former_residence_of-> **El Greco**

The respective general fundamental relationship "*Place has met Actor"* is:

E53.Place--(P89B.contains)$^{[0,n]}$ -> **E53.Place:**

{E53.Place -- P74B.is_current_or_former_residence_of-> **E39.Actor:**

{E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$->**E39.Actor** [-- P2F.has_type -> **E55.Type**]

}

OR

E53.Place --P7B.witnessed->**E5.Event:**

{E5.Event --(P9F.consists_of)$^{[0,n]}$-> **E5.Event:**

{E5.Event-- P12F.occurred_in_the_presence_of ->**E39.Actor:**

{E39.Actor --(P107B.is_current_or_former_member)$^{[0,n]}$-> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

}

}

Specializations:

The user frequently may only be interested in actors (individuals or groups) that originate from a place so a specialization of the former relationship is:

- is_origin_of

An example from real metadata would be:

***Place** is origin of* **El Greco**

Crete -- P88F.consists_of->**Fodele--** P7B.witnessed->**Birth of El Greco --** P98F.brought_into_life-> **El Greco**

OR

Fodele-- P7B.witnessed->**Birth of El Greco --**P98F.brought_into_life-> **El Greco**

The respective specialized fundamental relationship "*Place is origin of Actor*" is:

E53.Place--(P89B.contains)$^{[0,n]}$ -> **E53.Place:**

{E53.Place --P7B.witnessed->**E5.Event:**

{E5.Event --{(P9F.consists_of)$^{[0,n]}$ OR (P9B.forms_part_of)$^{[0,n]}$)-> **E5.Event:**

{E63.Beginning_of_Existence--P98F.brought_into_life->**E39.Actor:**

{E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$->

E39.Actor [--P2F.has_type -> **E55.Type**]

}

}

}

}

Place-Event

The Place-Event fundamental relationships can be switched to Thing-Time fundamental relationships, by further adding the CIDOC-CRM property P4F.has_time-span at the range category Event **(E5.Event** -- P4F.has_time-span->**E52.Time-Span)**. This happens because Time refers to the chronological definition of Events.

a. has met

This relationship is used to show in what Places an Event took Place. An event may take place in various places, if we consider that the sub-events possibly take place in other places. We consider that the main event occurs in all the places where its sub-events take place.

An example from real metadata would be:

Place** has met **Laser scanning acquisition of Canoe-shaped vase

Nicosia--P89B.contains **->ArchaeologicalMuseumofNicosia-ConservationLaboratory**--P7B.witnessed-> **Laser scanning acquisition of Canoe-shaped vase**

The respective specialized fundamental relationship "*Place has met Event"* is:

E53.Place-- (P89B.contains)$^{[0,n]}$ -> **E53.Place:**

{E53.Place--P7B.witnessed-> **E5.Event:**

{E5.Event -- (P9B.forms_part_of)$^{[0,n]}$->**E5.Event** [--P2F.has_type -> ***E55.Type****]*

}

}

b. refers to

A Place can refer to an Event through objects that have as theme or depict an Event and are or were located in the Place or through people that are at a Place and that mention or refer to an Event.

In CIDOC-CRM properties that define the "refer to" or "is about" relationship are :

- P62F.depicts
- P67F.refers_to

An example from real metadata would be:

Place** refers to **French Revolution

Paris--P89B.contains **->Louvre Museum** --P53B.is_former_or_current_location_of ->**"La Liberté guidant le peuple"--** P62F.depicts-- ***French Revolution***

The respective general fundamental relationship "*Place refers to Event"* is:

E53.Place--(P89B.contains)$^{[0,n]}$-> **E53.Place:**

{E53.Place-- P53B.is_former_or_current_location_of -> **E24.Physical_Man-Made_Thing:**

{ E24.Physical_Man-Made_Thing -- (P46F.is_composed_of) $^{[0,n]}$ **->** **E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- P62F.depicts-> **E5.Event**:

{ E5.Event --(P9F.consists_of)$^{[0,n]}$-> **E5.Event** [--P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing--P128F.carries->
E73.Information_Object:

{E73.Information_Object --P67F.refers_to -> **E5.Event**:

{E5.Event --(P9F.consists_of)$^{[0,n]}$-> **E5.Event** [--
P2F.has_type -> **E55.Type**]

}

}

}

}

d. is referred to at

A Place may be referred by Events during which Things that refer to the Place are created.

In CIDOC-CRM properties that define the "is referred to at" relationship are:

- P67B.is_referred_to_by
- P62B.is_depicted_by

An example from real metadata would be:

Place** is referred to at **Abraham Ortelius Cartography Event

Europe -- P62B.is_depicted_by -> **Europe Map --** P108B.was_produced_by-> **Abraham Ortelius Cartography Event**

The respective general fundamental relationship "*Place is referred to at Event"* is:

E53.Place--(P89F.falls_within)$^{[0,n]}$-> **E53.Place:**

{ E53.Place --P67B.is_referred_to_by -> **E89.Propositional_Object:**

{E89.Propositional_Object --{ (P130F.shows_features_of)$^{[0,n]}$ OR (P130B.features_are_also_found_on) $^{[0,n]}$ }-> **E89.Propositional_Object**:

{E89.Propositional_Object -- P94B.was_created_by -> **E65.Creation:**

{ E65.Creation -- (P9B.forms_part_of)$^{[0,n]}$ -> **E5.Event** [--P2F.has_type -> **E55.Type**]

}

OR

E73.Information_Object -- P128B.is_carried_by -> **E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) $^{[0,n]}$ -> **E24.Physical_Man-Made_Thing**:

{ E24.Physical_Man-Made_Thing--P108B.was_produced_by -> **E12.Production:**

{ E12.Production -- (P9B.forms_part_of)$^{[0,n]}$ -> **E5.Event** [--P2F.has_type -> **E55.Type**]

}

}

}

}

OR

E53.Place -- P62B.is_depicted_by-> **E24.Physical_Man-Made_Thing:**

{ **E24.Physical_Man-Made_Thing** --{ (P130F.shows_features_of)$^{[0,n]}$ OR (P130B.features_are_also_found_on) $^{[0,n]}$ }-> **E24.Physical_Man-Made_Thing**:

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) [0,n] ->
E24.Physical_Man-Made_Thing:

{E24.Physical_Man-Made_Thing--P108B.was_produced_by ->
E12.Production:

{ E12.Production -- (P9B.forms_part_of) [0,n] -> **E5.Event**
[--P2F.has_type -> **E55.Type**]

}

}

}

}

}

Place-Concept

a. has type

This relationship connects a place with its type. A place may belong to more than one type categories.

An example from real metadata would be:

Place** has type **room

ArchaeologicalMuseumofNicosia-ConservationLaboratory -- P2F.has_type->**room**

The respective fundamental relationship "*Place has type Concept"* is:

E53.Place -- P2F.has_type ->**E55.Type**:

{ **E55.Type** -- {(P127F.has_broader_term)$^{[0,n]}$ OR (P2F.has_type)$^{[0,n]}$}-> **E55.Type**
[P2F.has_type -> **E55.Type**]

}

ACTOR

The Actor category comprises people, either individually or in groups, who have the potential to perform intentional actions for which they can be held responsible. Since people may be members of groups and groups can be members of other groups, the following relations can be applied at the beginning of any of the fundamental relationships. So it is displayed here as root of the relationships path-chain.

E39.Actor--(P107B.is_current_or_former_member_of)[0,n]->

E74.Group

OR

E74.Group--(P107F.has_current_or_former_member)[0,n]->

E39.Actor

Actor-Place

a. refers to

An actor may refer to a Place by Things they create and refer to or have as subject a Place.

An example from real metadata would be:

Actor** refers to **Ithaca

Thomas Miles Richardson, Jnr. -- P14B.performed ->**"Peasants on the coast before the Island of Ithaca Painting Event" --** P108F.has_produced ->**"Peasants on the coast before the Island of Ithaca"** -- P62F.depicts ->**Ithaca**

OR

K.P. Kavafis -- P14B.performed ->"**Ithaca poem writing**" -- P94F.has_created ->**"Ithaka" --** P67F.refers_to ->**Ithaka**

The respective fundamental relationship "*Actor refers to Place"* is:

E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$->**E39.Actor:**

{E39.Actor -- P14B.performed -> **E7.Activity:**

{E7.Activity -- (P9F.consists_of)$^{[0,n]}$-> **E5.Event:**

{E7.Activity--P92F.brought_into_existence **-> C1.Object:**

{**C1.Object** --{ (P130F.shows_features_of)$^{[0,n]}$ OR (P130B.features_are_also_found_on) $^{[0,n]}$ }-> **C1.Object**:

{E24.Physical_Man-Made_Thing -- P62F.depicts **->** **E53.Place:**

{E53.Place --(P89B.contains**)$^{[0,n]}$->** **E53.Place** [--P2F.has_type -> **E55.Type**] }

OR

E89.Propositional_Object --P67F.refers_to-> **E53.Place:**

{E53.Place --(P89B.contains**)$^{[0,n]}$->** **E53.Place** [--P2F.has_type -> **E55.Type**] }

OR

E24.Physical_Man-Made_Thing -- P128F.carries -> **E73.Information_Object:**

{E73.Information_Object -- P67F.refers_to -> **E53.Place:**

{E53.Place --(P89B.contains**)$^{[0,n]}$->** **E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

}

}

}

}

}

b. is referred to at

There are cases that one is interested in Actors that are referred to at a certain Place. Reference implies the presence or creation of a Thing that refers to the Actor in interest, as even speech is a human product, thus a Thing. So, when we talk about something, write about something or in any other way mention something the means we do it is mapped to a Thing that we create.
In this manner we connect the Thing and Place FCs not only directly but also through other Things and through Events.

In CIDOC-CRM properties that define the "is referred to at" relationship are:

- P67B.is_referred_to_by
- P62B.is_depicted_by

An example from real metadata would be:

Actor** is referred to at **Crete

"Prince of the Lilies" -- P67B.is_referred_to_by**->"Wall paintings on Knossos Palace" --** P128B.is_carried_by **->"Knossos Palace walls" --** P53F.has_former_or_current_location -> **Knossos--** P89F.falls_within **->Crete**

The respective general fundamental relationship "*Actor is referred to at Place"* is:

E39.Actor--(P107B.is_current_or_former_member_of)$^{[0,n]}$-> **E39.Actor:**

{E39.Actor -- P67B.is_referred_to_by -> **E89.Propositional_Object:**

{E73.Information_Object -- P94B.was_created_by -> **E65.Creation**:

{E65.Creation --(P9B.forms_part_of)$^{[0,n]}$ ->**E5.Event:**

{E5.Event -- P7F.took_place_at ->**E53.Place:**

{E53.Place --(P89F.falls_within)**$^{[0,n]}$-> E53.Place**

[--P2F.has_type -> **E55.Type**]

}

}

}

OR

E73.Information_Object -- P128B.is_carried_by -> **E24.Physical_Man-Made_Thing:**

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) $^{0,n)}$ -> **E18.Physical_Thing**:

{**E18.Physical_Thing --** P53F.has_former_or_current_location -> **E53.Place :**

{E53.Place --(P89F.falls_within)**$^{[0,n]}$-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

}

}

OR

E39.Actor -- P62B.is_depicted_by **-> E24.Physical_Man-Made_Thing:**

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) $^{0,n)}$ ->
E24.Physical_Man-Made_Thing:

{ **E24.Physical_Man-Made_Thing --**
P53F.has_former_or_current_location -> **E53.Place:**

{E53.Place --(P89F.falls_within**)**$^{[0,n]}$**-> E53.Place**

[--P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P108B.was_produced_by ->
E12.Production:

{ E12.Production --(P9B.forms_part_of)$^{[0,n]}$-> **E5.Event:**

{ E5.Event -- P7F.took_place_at ->**E53.Place**:

{E53.Place --(P89F.falls_within**)**$^{[0,n]}$**-> E53.Place** [--
P2F.has_type -> **E55.Type**]

}

}

}

}

}

}

c. has met

This is a general relationship that connects an Actor with a place they have been at. This can also happen through an event.

An example from real metadata would be:

Actor** has met **Crete

Sir Arthour Evans -- P12B.was_present_at ->"**Excavations at Knossos**"-> P7F.took_place_at->"**Knossos**" -- P88B.forms_part_of->**Crete**

OR

Eleutherios Venizelos -- P74F.has_current_or_former_residence->**Chania**-> P88B.forms_part_of->**Crete**

The respective fundamental relationship "*Actor has met Place"* is:

E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$-> **E39.Actor:**

{E39.Actor--P12B.was_present_at-> **E5.Event:**

{E5.Event--(P9B.forms_part_of)$^{[0,n]}$-> **E5.Event:**

{E5.Event --P7F.took_place_at-> **E53.Place**:

{E53.Place --(P89F.falls_within**)$^{[0,n]}$-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

}

OR

E39.Actor -- P74F.has_current_or_former_residence **-> E53.Place**:

{E53.Place --(P89F.falls_within**)$^{[0,n]}$-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

d. from

An actor is from a place usually when their residence is located there, or they have been born there.

An example from real metadata would be:

Actor** from **Crete

Kazantzakis -- P98B.was_born ->"**Kazantzakis birth**"-> P7F.took_place_at->"**Mirtia**" -- P88B.forms_part_of->**Crete**

OR

Eleutherios Venizelos -- P74F.has_current_or_former_residence->**Chania**-> P88B.forms_part_of->**Crete**

The respective fundamental relationship "*Actor from Place"* is:

E39.Actor--(P107B.is_current_or_former_member_of)$^{[0,n]}$-> **E39.Actor:**

{E39.Actor-- P74F.has_current_or_former_residence-> **E53.Place**:

{E53.Place --(P89F.falls_within**)$^{[0,n]}$-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

OR

E39.Actor -- P92B.was_brought_into_existence_by -> **E63.Beginning_of_Existence**:

{E63.Beginning_of_Existence --(P9B.forms_part_of)[0,n]-> **E5.Event:**

{E5.Event --P7F.took_place_at->**E53.Place**:

{E53.Place --(P89F.falls_within**)[0,n]-> E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

}

}

Actor-Thing

a. refers to

An actor may refer to a Thing by Things they create and refer to or have as subject a Thing.

An example from real metadata would be:

Actor** refers to **Notre Dame

Edouard Cortes-- P14B.performed ->**"Notre Dame painting Event" --** P108F.has_produced -> **"Notre Dame painting"** -- P62F.depicts ->**Notre Dame**

OR

Kiran Keswani -- P14B.performed ->" **THE NOTRE DAME CATHEDRAL poem writing**" -- P94F.has_created ->**" THE NOTRE DAME CATHEDRAL" --** P67F.refers_to -> **Notre Dame**

The respective fundamental relationship "*Actor refers to Thing"* is:

E39.Actor --(P107F.has_current_or_former_member)[0,n]->**E39.Actor:**

{**E39.Actor** -- P14B.performed -> **E7.Activity:**

{**E7.Activity** -- (P9F.consists_of)[0,n]-> **E7.Activity:**

{**E7.Activity**--P92F.brought_into_existence -> **C1.Object:**

{**E24.Physical_Man-Made_Thing** --
{(P130F.shows_features_of)[0,n] OR
(P130B.features_are_also_found_on) [0,n]}->
E24.Physical_Man-Made_Thing:

{**E24.Physical_Man-Made_Thing** -- P62F.depicts ->
C1.Object:

{**C1.Object** -- { (P46F.is_composed_of) [0,n] OR
(P106F.is_composed_of) [0,n] OR
(P148F.has_component) [0,n]} -> **C1.Object** [--
P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P128F.carries ->
E73.Information_Object:

{**E73.Information_Object** -- P67F.refers_to ->
C1.Object:

{**C1.Object** -- { (P46F.is_composed_of) [0,n]
OR (P106F.is_composed_of) [0,n] OR
(P148F.has_component) [0,n]} -> **C1.Object** [
-- P2F.has_type -> **E55.Type**]

}

}

}

}

}

}

}

b. is referred to by

An Actor may be referred to by Things that have as theme or subject the Actor, or that refer to or are about the Actor.

In CIDOC-CRM properties that define the "is referred to by" relationship are:

- P67B.is_referred_to_by
- P62B.is_depicted_by

An example from real metadata would be:

Actor** is referred to by "**Thetis Accepting the Shield of Achilles from Vulcan" Painting

Thetis -- P67B.is_referred_to_by ->"**Thetis Accepting the Shield of Achilles from Vulcan" Painting**

The respective general fundamental relationship "*Actor is referred to by Thing"* is:

E39.Actor -- (P107B.is_current_or_former_member_of)$^{[0,n]}$ -> **E39.Actor:**

{E39.Actor --P67B.is_referred_to_by -> **E89.Propositional_Object:**

{E89.Propositional_Object -- (P148B.is_component_of)$^{[0,n]}$ ->

E89.Propositional_Object:

{ E89.Propositional_Object[--P2F.has_type -> **E55.Type**]

OR

E73.Information_Object -- P128B.is_carried_by ->
E24.Physical_Man-Made_Thing:

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) $^{0,n)}$
-> **E24.Physical_Man-Made_Thing**[--P2F.has_type ->
E55.Type]

}

}

}

OR

E39.Actor -- P62B.is_depicted_by-> **E24.Physical_Man-Made_Thing**:

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) $^{0,n)}$ ->
E24.Physical_Man-Made_Thing[--P2F.has_type -> **E55.Type**]

}

}

c. is origin of

Things are created by actors and are kept or owned by them. So, actors can serve as creators or owners of Things. These properties can be unified into one fundamental relationship that shows the origin of a Thing. This is the reverse relation of *Thing from Actor*.

The respective fundamental relationship "*Actor is origin of Thing"* is:

E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$->**E39.Actor*:***

{E39.Actor -- P14B.performed -> **E7.Activity:**

{E7.Activity -- (P9F.consists_of)[0,n]-> **E5.Event:**

{E63.Beginning_of_Existence --P92F.brought_into_existence **->**
C1.Object:

{C1.Object -- { (P46F.is_composed_of) [0,n] OR
(P106F.is_composed_of) [0,n] OR (P148F.has_component)
[0,n]}-> **C1.Object** [-- P2F.has_type -> **E55.Type**]

}

}

}

OR

E39.Actor --{ P49B.is_former_or_current_keeper_of OR
P51B.is_former_or_current_owner_of }-> **E18.Physical_Thing:**

{E18.Physical_Thing -- (P46F.is_composed_of) [0,n] ->

E18.Physical_Thing [--P2F.has_type -> **E55.Type**]

}

}

Specializations:

In the "*is origin of*" FR we can define specializations to distinguish between the creator of one Thing and the owner /keeper of it. So, we have:

- is generator of

An example from real metadata would be:

Actor** is generator of **Statue of Zeus at Olympia

Feidias-- P14B.performed ->"**Sculpturing of statue of Zeus at Olympia**" -- P108F.has_produced -> **Statue of Zeus at Olympia**

The respective fundamental relationship "*Actor is generator of Thing"* is:

E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$->**E39.Actor*:***

{E39.Actor -- P14B.performed -> **E7.Activity:**

{E7.Activity -- (P9F.consists_of)$^{[0,n]}$-> **E5.Event:**

{E63.Beginning_of_Existence --P92F.brought_into_existence **->**

C1.Object:

{C1.Object -- { (P46F.is_composed_of) $^{[0,n]}$ OR (P106F.is_composed_of) $^{[0,n]}$ OR (P148F.has_component) $^{[0,n]}$}-> **C1.Object** [-- P2F.has_type -> **E55.Type**]

}

}

}

}

- has

An actor can have a Thing under their possession or keep.

An example from real metadata would be:

Actor** has **Kazantakis' letters to Aggelos Sikelianos

Anna Sikelianou -- P49B.is_former_or_current_keeper_of -> **Kazantakis' letters to Aggelos Sikelianos**

OR

Nikos Kazantzakis Museum Foundation -- P52B.is_current_owner_of -> **Kazantakis' letters to Aggelos Sikelianos**

The respective fundamental relationship "*Actor has Thing"* is:

E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$->**E39.Actor*:***

{E39.Actor --{ P49B.is_former_or_current_keeper_of OR P51B.is_former_or_current_owner_of }-> **E18.Physical_Thing:**

{E18.Physical_Thing -- (P46F.is_composed_of) $^{[0,n]}$ ->

E18.Physical_Thing [--P2F.has_type -> **E55.Type**]

}

}

d. has met

This is a general relationship connecting an Actor with Things that they have met (been at the same place the same time) at least once. This is possible when the actor and the Thing were present at the same Event. Nevertheless a Thing may be documented to be present to a certain Event and so may be an Actor, but this may not necessarily mark that the two entities have met each other. This may happen when we are talking about a big event, such as the French Revolution, that takes place in various areas in the same or different time sub-

periods. However, since here we are interested in high recall rates we prefer to include such uncertainties than to exclude valid cases.

An example from real metadata would be:

***Actor** has met* **Arkadi Monastery**

Mustafa Pasha --P12B.was_present_at->"**Arkadi Monastery Explosion**" --P12F.occurred_in_the_presence_of-> **Arkadi Monastery**

The respective fundamental relationship "*Actor has met Thing"* is:

E39.Actor--P12B.was_present_at-> **E5.Event*:***

{ E5.Event -- (P9F.consists_of)$^{[0,n]}$-> **E5.Event:**

{E5.Event-- P12F.occurred_in_the_presence_of ->

E70.Thing[--P2F.has_type -> **E55.Type**]

}

}

Actor-Actor

a. has met

This is a general relationship connecting an Actor with Actors that they have met (been at the same place the same time) at least once. This is possible when the actors were present at the same Event. Nevertheless two actors may be documented to be present to a certain Event, but this may not necessarily mark that the two entities have met each other. This may happen when we are talking about a big event, such as the French Revolution, that takes place in various areas in the same or different time sub-periods. However, since here we are interested in high recall rates we prefer to include such uncertainties than to exclude valid cases.

An example from real metadata would be:

***Actor** has met* **Mustafa Pasha**

Hegumen Gavriil--P12B.was_present_at->"**Arkadi Monastery Explosion**" --P12F.occurred_in_the_presence_of-> **Mustafa Pasha**

The respective fundamental relationship "*Actor has met Actor"* is:

E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$-> **E39.Actor:**

{E39.Actor--P12B.was_present_at-> **E5.Event:**

{E5.Event-- (P9F.consists_of)$^{[0,n]}$ -> **E5.Event:**

{E5.Event --P12F.occurred_in_the_presence_of-> **E39.Actor:**

{ E39.Actor--(P107B.is_current_or_former_member_of)$^{[0,n]}$->
E39.Actor[--P2F.has_type -> **E55.Type**]

}

}

}

}

b. refers to

An actor can refer to another actor by Things that they create and refer to another Actor.

An example from real metadata would be:

***Actor** refers to* **Alexander the Great**

Charles Le Brun -- P14B.performed ->**" Le Brun, Alexander and Porus painting Event" --** P108F.has_produced ->**" Le Brun, Alexander and Porus painting"->** P62F.depicts -> **Alexander the Great**

OR

Lysippos -- P14B.performed ->**"Bust of Alexander the Great Copy Creation"--** P108F.has_produced --**"Bust of Alexander the Great Copy "->**P130F.shows_features_of---- **"Bust of Alexander the Great**"-- P128F.carries **->"Inscription of Alexander the Great Bust"--** P129F.is_about-> **Alexander the Great**

The respective fundamental relationship "*Actor refers to Actor"* is:

E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$-> **E39.Actor:**

{E39.Actor -- P14B.performed -> **E7.Activity:**

{E7.Activity -- (P9F.consists_of)$^{[0,n]}$-> **E5.Event:**

{E63.Beginning_of_Existence --{ P92F.brought_into_existence **} -> C1.Object:**

{**C1.Object--** { (P130F.shows_features_of)$^{[0,n]}$ OR (P130B.features_are_also_found_on) $^{[0,n]}$ }-> **C1.Object**:

{E24.Physical_Man-Made_Thing -- P62F.depicts **->**

E39.Actor:

{E39.Actor--

(P107F.has_current_or_former_member)$^{[0,n]}$

-> **E39.Actor**[--P2F.has_type -> **E55.Type**]

}

OR

E89.Propositional_Object --P67F.refers_to->
E39.Actor:

{E39.Actor--
(P107F.has_current_or_former_member)$^{[0,n]}$-
> **E39.Actor**[--P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P128F.carries ->
E73.Information_Object:

{E73.Information_Object --P67F.refers_to
-> **E39.Actor:**

{E39.Actor--
(P107F.has_current_or_former_me
mber)$^{[0,n]}$-> **E39.Actor**[--
P2F.has_type -> **E55.Type**]

}

}

}

}

}

}

c. is referred to by

An Actor may be referred to by Actors though artifacts that they create and have as a theme or subject or refer to the Actor.

In CIDOC-CRM properties that define the "is referred to by" relationship are:

- P67B.is_referred_to_by
- P62B.is_depicted_by

An example from real metadata would be:

Actor** is referred to by **Kazantzakis

Alexis Zorbas -- P129B.is_subject_of ->**"Alexis Zorbas Book"**-- P94B.was_created_by -> "**Alexis Zorbas Book Writing**" -- P14F.carried_out_by-->**Kazantzakis**

The respective general fundamental relationship "*Actor is referred to by Actor"* is:

E39.Actor -- (P107B.is_current_or_former_member_of)$^{[0,n]}$ -> **E39.Actor:**

{E39.Actor --P67B.is_referred_to_by-> **E89.Propositional_Object:**

{E89.Propositional_Object -- P94B.was_created_by ->**E65.Creation**:

{E65.Creation -- (P9B.forms_part_of)$^{[0,n]}$-> **E7.Activity:**

{E7.Activity -- P14F.carried_out_by-> **E39.Actor:**

{E39.Actor --
(P107B.is_current_or_former_member_of) $^{[0,n]}$
->**E39.Actor** [--P2F.has_type ->**E55.Type**]

}

}

}

OR

E73.Information_Object -- P128B.is_carried_by -> **E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) $^{0,n)}$ -> **E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- P108B.was_produced _by **-> E12.Production:**

{ E12.Production -- (P9B.forms_part_of)$^{[0,n]}$-> **E7.Activity:**

{E7.Activity -- P14F.carried_out_by-> **E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of) $^{[0,n]}$ ->**E39.Actor** [-- P2F.has_type ->**E55.Type**]

}

}

}

}

}

OR

E39.Actor -- P62B.is_depicted_by-> **E24.Physical_Man-Made_Thing**:

{E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) $^{[0,n]}$ **->** **E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- P108B.was_produced _by **->** **E12.Production:**

{ E12.Production -- (P9B.forms_part_of)[0,n]-> **E7.Activity:**

{E7.Activity -- P14F.carried_out_by-> **E39.Actor:**

{E39.Actor --

(P107B.is_current_or_former_member_of)

[0,n] ->**E39.Actor** [--P2F.has_type ->**E55.Type**]

}

}

}

}

}

}

d. from

This relationship connects an Actor with their generators. Generator may mean parent if we are talking about individuals or founder if we are talking about groups of people.

An example from real metadata would be:

Actor** from **Nikolaos Skoufas

Filiki Etaireia -- P95B.was_formed_by ->'**Filiki Etaireia foundation Event**'-- P14F.carried_out_by ->Nikolaos Skoufas

The respective fundamental relationship "*Actor from Actor*" is:

E39.Actor -- (P107B.is_current_or_former_member_of)$^{[0,n]}$ -> **E39.Actor:**

{E21.Person--P98B.was_born **-> E67.Birth:**

{E67.Birth--{P97F.from_father OR P96F.by_mother**}->E21.Person**[--P2F.has_type -> **E55.Type**]

}

OR

E74.Group -- P95B.was_formed_by **-> E66.Formation :**

{E66.Formation-- (P9F.consists_of)$^{[0,n]}$-> **E7.Activity:**

{ E7.Activity--P14F.carried_out_by **E39.Actor:**

{E39.Actor --(P107B.is_current_or_former_member_of) $^{[0,n]}$ -> **E39.Actor** [--P2F.has_type ->**E55.Type**]

}

}

}

}

e. is origin of

This is the reverse relationship of from and as such it returns the children or the groups one Actor has created.

An example from real metadata would be:

Actor** is origin of **Filiki Etaireia

Nikolaos Skoufas -- P14B.performed ->'**Filiki Etaireia foundation Event**'-- P95F.has_formed -> **Filiki Etaireia**

The respective fundamental relationship "*Actor is oringin of Actor"* is:

E39.Actor-- (P107F.has_current_or_former_member)$^{[0,n]}$-> **E39.Actor:**

{**E21.Person** -- {P97B.was_father_for OR P96B.gave_birth }-> **E67.Birth:**

{E67.Birth--P98F.brought_into_life**->E21.Person**[--P2F.has_type -> **E55.Type**]

}

OR

E39.Actor --P14B.performed -> **E7.Activity**:

{**E7.Activity --** (P9B.forms_part_of)$^{[0,n]}$-> **E5.Event:**

{**E66.Formation** -- P95F.has_formed -> **E39.Actor:**

{E39.Actor-- (P107F.has_current_or_former_member)$^{[0,n]}$-> **E39.Actor**[--P2F.has_type -> **E55.Type**]

}

}

}

}

f. has member

With this FR the groups to which an actor belongs to are returned. This relationship between groups and actors can be documented either explicitly or through a Joining Event,

An example from real metadata would be:

Actor** has member **Theodoros Kolokotronis

Filiki Etaireia -- P107F.has_current_or_former_member **->Theodoros Kolokotronis**

The respective fundamental relationship "*Actor has member Actor"* is:

E74.Group--(P107F.has_current_or_former_member)$^{[0,n]}$-> **E39.Actor :**

{E39.Actor[--P2F.has_type -> **E55.Type**]

OR

E74.Group --P144B.gained_member_by -> **E85.Joining:**

{E85.Joining --P143B.joined -> **E39.Actor:**

{E39.Actor-- (P107F.has_current_or_former_member)$^{[0,n]}$->
E39.Actor[--P2F.has_type -> **E55.Type**]

}

}

}

g. is member of

This is the reverse link of the *has member* relationship. It returns the members of a certain group.

An example from real metadata would be:

Actor** is member of **Filiki Etaireia

Theodoros Kolokotrwnis -- P107B.is_current_or_former_member_of-> **Filiki Etaireia**

OR

Odysseas Androutsos -- P143B.was_joined_by ->"**Joining of Androutsos to Filiki Etaireia**" -- P144F.joined_with -> **Filiki Etaireia**

The respective fundamental relationship "*Actor is member of Actor"* is:

E39.Actor-- **(**P107B.is_current_or_former_member_of)$^{[0,n]}$-> **E39.Actor :**

{**E74.Group** [--P2F.has_type -> **E55.Type**]

OR

E39.Actor -- **P143B.was_joined_by -> **E85.Joining:

{ E85.Joining -- P144F.joined_with -> **E39.Actor:**

{E39.Actor -- **(**P107B.is_current_or_former_member_of)$^{[0,n]}$->

E39.Actor[-- P2F.has_type -> **E55.Type**]

}

}

}

Actor-Event

The Actor-Event fundamental relationships can be switched to Actor-Time fundamental relationships, by further adding the CIDOC-CRM property P4F.has_time-span at the range category Event **(E5.Event** --P4F.has_time-span->**E52.Time-Span)**. This happens because Time refers to the chronological definition of Events.

a. refers to

An actor may refer to an Event through Things they produce or create and which Things have as subject or refer to the Event.

An example from real metadata would be:

Actor *refers to* ***Trojan War***

Sosias-- P14B.performed **->" Achilles tending Patroclus wounded by an arrow painting Event" --** P108F.has_produced -> **" Achilles tending Patroclus wounded by an arrow painting"** -- P128F.carries -> **"Inscription "** -> P67F.refers_to ->**Trojan War**

The respective fundamental relationship "*Actor refers to Event"* is:

E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$-> **E39.Actor:**

{E39.Actor -- P14B.performed -> **E7.Activity:**

{E7.Activity -- (P9F.consists_of)$^{[0,n]}$-> **E5.Event:**

{E63.Beginning_of_Existence --P92F.brought_into_existence **-> C1.Object:**

{C1.Object -- { (P46F.is_composed_of) $^{[0,n]}$ OR (P148F.has_component) $^{[0,n]}$}-> **C1.Object:**

{**C1.Object--** { (P130F.shows_features_of)$^{[0,n]}$ OR (P130B.features_are_also_found_on) $^{[0,n]}$ }-> **C1.Object:**

{E24.Physical_Man-Made_Thing -- P62F.depicts **-> E5.Event:**

{E5.Event-- (P9F.consists_of)$^{[0,n]}$->

E5.Event [-- P2F.has_type -> **E55.Type**]

}

OR

E89.Propositional_Object --P67F.refers_to -> **E5.Event:**

{E5.Event-- (P9F.consists_of)$^{[0,n]}$->

E5.Event [-- P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P128F.carries -> **E73.Information_Object:**

{E73.Information_Object -- P67F.refers_to -> **E5.Event:**

{E5.Event-- (P9F.consists_of)$^{[0,n]}$->

E5.Event [-- P2F.has_type ->

E55.Type]

}

}

}

}

}

}

}

b. is referred to at

An Actor may be referred by Events during which Things that refer to the Actor are created.

In CIDOC-CRM properties that define the “is referred to at” relationship are:

- P67B.is_referred_to_by
- P62B.is_depicted_by

An example from real metadata would be:

Actor** is referred to at **Las Meninas painting Event

Velazquez-- P62B.is_depicted_by -> "**Las Meninas" painting --** P108B.was_produced_by-> **Las Meninas painting Event**

The respective general fundamental relationship "*Actor is referred to at Event"* is:

E39.Actor -- (P107B.is_current_or_former_member_of)$^{[0,n]}$ -> **E39.Actor:**

{E39.Actor -- P67B.is_referred_to_by -> **E89.Propositional_Object:**

{E89.Propositional_Object -- P94B.was_created_by -> **E65.Creation:**

{E65.Creation -- (P9B.forms_part_of) $^{[0,n]}$ -> **E5.Event** [--P2F.has_type -> **E55.Type**]

}

OR

E73.Information_Object -- P128B.is_carried_by -> **E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) $^{[0,n]}$ -> **E24.Physical_Man-Made_Thing**:

{E24.Physical_Man-Made_Thing--P108B.was_produced_by -> **E12.Production:**

{E12.Production -- (P9B.forms_part_of) $^{[0,n]}$ -> **E5.Event** [--P2F.has_type -> **E55.Type**]

}

}

}

}

OR

E39.Actor -- P62B.is_depicted_by-> **E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) [0,n] ->
E24.Physical_Man-Made_Thing:

{E24.Physical_Man-Made_Thing--P108B.was_produced_by ->
E12.Production:

{E12.Production -- (P9B.forms_part_of) [0,n] -> **E5.Event** [--
P2F.has_type -> **E55.Type**]

}

}

}

}

c. from

This relationship returns all the Actors that are brought into existence, so have origin in a specific Event. We assume a group and the member-groups are formed concurrently. Here we have to distinguish the cases between groups and persons as it would be a mistake that would diminish precision if we would say that individual members of groups were brought into existence at the same event as the group's formation. Eg John that is a member of FORTH institute was not born the same day as FORTH's establishment.

An example from real metadata would be:

Actor** from **Greek War of Independence

Filiki Etaireia -- P92B.was_brought_into_existence_by **-> "Filiki Etaireia Foundation"** -- P9B.forms_part_of -> **Greek War of Independence**

The respective specialized fundamental relationship "*Actor from Event*" is:

{E74.Group--(P107B.is_current_or_former_member)$^{[0,n]}$-> **E74.Group:**

{E74.Group --P95.was_formed_by ->E66.Formation:

{E66.Formation --(P9B.forms_part_of)$^{[0,n]}$ ->**E5.Event** [--P2F.has_type -> **E55.Type**]

}

}

OR

E21.Person -- P98B.was_born ->E67.Birth:

{ E67.Birth --(P9B.forms_part_of)$^{[0,n]}$ ->**E5.Event** [--P2F.has_type -> **E55.Type**]

}

}

d. has met

This is a general relationship that connects an Actor with an event they have been present at. It does not specify the role of the Actor in the Event. For more specification specializations on this FR are provided afterwards.

An example from real metadata would be:

Actor** has met **Trojan War

Patroklos -- P12B.was_present_at -> **Trojan War**

The respective fundamental relationship "*Actor has met Event"* is:

E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$-> **E39.Actor:**

{E39.Actor--P12B.was_present_at-> **E5.Event:**

{E5.Event--(P9B.forms_part_of)$^{[0,n]}$ ->**E5.Event** [--P2F.has_type -> **E55.Type**]

}

}

Specializations:

In the fundamental relationship *has met* we can also define some sub-fundamental relationships, which contain more restricted information. This information is however commonly asked for by the users and a more specialized query may return better results precision-wisely based to what they actually want to know.

- Performed action at

This relationship returns all the Actors that performed some kind of action in an Event.

An example from real metadata would be:

Actor** performed action at **Parthenon Construction

Phidias -- P14B.performed -> "**Parthenon Frieze Creation" --** P9B.forms_part_of -> **Parthenon Construction**

The respective specialized fundamental relationship "*Actor performed action at Event"* is:

E39.Actor --(P107F.has_current_or_former_member)$^{[0,n]}$-> **E39.Actor:**

{E39.Actor-- P14B.performed->**E7.Activity:**

{ **E7.Activity--(**P9B.forms_part_of)[0,n] ->**E5.Event** [--P2F.has_type -> **E55.Type**]

}

}

Actor-Concept

e. has type

This relationship connects an Actor with its type, which describes its concept. An Actor may belong to more than one type categories. Also by including the types of the members, we actually mean that the wider Actor "includes" the members' types.

An example from real metadata would be:

Actor** has type **Museum

Historical Museum of Crete -- P2F.has_type->**Museum**

The respective fundamental relationship "*Actor has type Concept"* is:

E39.Actor --(P107F.has_current_or_former_member)[0,n]-> **E39.Actor:**

{E39.Actor-- P2F.has_type-> **E55.Type**:

{ **E55.Type** -- {(P127F.has_broader_term)[0,n] OR (P2F.has_type)[0,n]}-> **E55.Type** [P2F.has_type -> **E55.Type**]

}

}

EVENT-TIME

The Event fundamental Category is indeed an important one. In the context of an event, actors, things, places and other events or time periods are included answering the 4 main questions WHO, WHEN, WHERE, WHAT. So, as the metadata of Events are potentially rich in information also regarding the other FCs, it becomes a considerable "link" between different FCs. But not only is it used for intermediate linking of Categories, but as a domain or range Category of one FR as well. In this set of FRs the Event category is used as Domain.

Time can be interpreted as the placement of the Event in Time and as such it can be used in the same way as the Event FC, only by adding the *is time-span of* property before the Event domain Category. That is:

E52.Time-Span-- P4B.is_time-span_of ->**E5.Event**

Event-Place

a. refers to

An event may refer to a Place by Things that are created or produced during this Event and refer to or depict the Place. A simple existence in an Event of a Thing that refers to a Place can not deduct that the Event also refers to the Place. Moreover we could say that an Event refers to the Place where this event took place, such as with the French Revolution that refers to France and also took place in France.

An example from real metadata would be:

Event** refers to **France

Trocadero Sunset painting Event -- P108F.has_produced -> **Trocadero Sunset painting** -- P62F.depicts ->**Paris** -- P89F.falls_within -> **France**

The respective general fundamental relationship "*Event refers to Place"* is:

E5.Event -- (P9F.consists_of)$^{[0,n]}$-> **E5.Event:**

{E63.Beginning_of_Existence --P92F.brought_into_existence **-> C1.Object:**

{**C1.Object**-- { (P130F.shows_features_of)$^{[0,n]}$ OR (P130B.features_are_also_found_on) $^{[0,n]}$ }-> **C1.Object**:

{C1.Object -- { (P46F.is_composed_of) $^{[0,n]}$ OR (P106F.is_composed_of) $^{[0,n]}$ OR (P148F.has_component) $^{[0,n]}$}-> **C1.Object:**

{E24.Physical_Man-Made_Thing -- P62F.depicts **->**

E53.Place:

{E53.Place --(P89B.contains)$^{[0,n]}$ -> **E53.Place** [--P2F.has_type -> **E55.Type**]

}

OR

E89.Propositional_Object --P67F.refers_to->

E53.Place:

{E53.Place --(P89B.contains)$^{[0,n]}$ -> **E53.Place** [--P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P128F.carries -> **E73.Information_Object:**

{E73.Information_Object -- P67F.refers_to ->

E53.Place:

{E53.Place --(P89B.contains) $^{[0,n]}$ -> **E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

}

}

}

}

b. is referred to at

There are cases that one is interested in Events that are referred to at a certain Place. Reference implies the presence or creation of a Thing that refers to the Event in interest, as even speech is a human product, thus a Thing. So, when we talk about something, write about something or in any other way mention something the means we do it is mapped to a Thing that we create.
In this manner we connect the Event and Place FCs not only directly but also through other Things and through Events.

In CIDOC-CRM properties that define the "is referred to" relationship are:

- P67B.is_referred_to_by
- P62B.is_depicted_by

An example from real metadata would be:

Event** is referred to at **Crete

"Trojan War" -- P67B.is_referred_to_by**->"Iliad" --** P128B.is_carried_by **->"Iliad Book ISBN 3458" --** P53F.has_former_or_current_location ->**Heraklion--** P89F.falls_within **->Crete**

The respective general fundamental relationship "*Actor is referred to at Place"* is:

E5.Event -- (P9B.forms_part_of)$^{[0,n]}$-> **E5.Event:**

{**E5.Event** -- P67B.is_referred_to_by -> **E89.Propositional_Object:**

{E89.Propositional_Object -- (P148B.is_component_of)$^{[0,n]}$->
E89.Propositional_Object:

{E73.Information_Object -- P94B.was_created_by -> **E65.Creation**:

{E65.Creation --(P9B.forms_part_of)$^{[0,n]}$-> **E5.Event**:

{E5.Event-- P7F.took_place_at ->**E53.Place:**

{E53.Place --(P89F.falls_within**)$^{[0,n]}$-> E53.Place**

[--P2F.has_type -> **E55.Type**]

}

}

}

OR

E73.Information_Object -- P128B.is_carried_by ->
E24.Physical_Man-Made_Thing:

{E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) $^{0,n)}$
-> **E24.Physical_Man-Made_Thing**:

{**E24.Physical_Man-Made_Thing --**
P53F.has_former_or_current_location -> **E53.Place :**

{E53.Place --(P89F.falls_within**)$^{[0,n]}$-> E53.Place** [-
-P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing --
P108B.was_produced_by -> **E12.Production**:

{ E12.Production --P9B.forms_part_of->
E5.Event:

{ E5.Event -- P7F.took_place_at ->

E53.Place:

{E53.Place --(P89F.falls_within**)**$^{[0,n]}$

-> E53.Place [--P2F.has_type ->

E55.Type]

}

}

}

}

}

}

OR

E5.Event -- P62B.is_depicted_by **-> E24.Physical_Man-Made_Thing:**

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of)$^{0,n)}$ ->

E24.Physical_Man-Made_Thing:

{ **E24.Physical_Man-Made_Thing --**

P53F.has_former_or_current_location -> **E53.Place:**

{E53.Place --(P89F.falls_within**)**$^{[0,n]}$**-> E53.Place**

[--P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P108B.was_produced_by ->

E12.Production:

{ E12.Production --P9B.forms_part_of-> **E5.Event:**

{ E5.Event -- P7F.took_place_at ->**E53.Place**:

{E53.Place --(P89F.falls_within**)**$^{[0,n]}$**->** **E53.Place** [--P2F.has_type -> **E55.Type**]

}

}

}

}

}

c. from

Events usually take place at one or more Places, which property is either directly connected to the Event, or may be inherited from super-events. If both cases are valid, we are interested in all the Places where an event and its super-events have taken place at.

An example from real metadata would be:

Event from Crete

Found of Knossos Palace -- P9B.forms_part_of ->**Knossos Excavations** -- P89F.falls_within-> **Crete**

The respective general fundamental relationship "*Event from Place"* is:

E5.Event-- (P9B.forms_part_of)$^{[0,n]}$ -> **E5.Event:**

{E5.Event--P7F.took_place_at-> **E53.Place**:

{E53.Place--(P89F.falls_within) $^{[0,n]}$ ->

E53.Place [--P2F.has_type -> **E55.Type**]

}

}

Event-Thing

a. refers to or is about

An event may refer to a Thing by Things that are created or produced in the Event and refer to or are about the Thing. In another aspect, an Event also refers to Things that are created, produced or destroyed in the Event, such as the Destruction of Parthenon that refers to Parthenon.

An example from real metadata would be:

Event** refers to or is about **Parthenon

Digitization Event of Acropolis -- P9F.consists_of -> **Capturing of the Parthenon Event --** P94F.has_created -> **The Parthenon.jpg** -- P129F.is_about -> **Parthenon**

The respective general fundamental relationship "*Event refers to or is about Thing"* is:

E5.Event -- (P9F.consists_of)$^{[0,n]}$-> **E5.Event:**

{E63.Beginning_of_Existence --P92F.brought_into_existence ->**C1.Object:**

{C1.Object --{(P46F.is_composed_of) $^{[0,n]}$ OR (P106F.is_composed_of) $^{[0,n]}$ OR (P148F.has_component) $^{[0,n]}$}-> **C1.Object:**

{**C1.Object--** { (P130F.shows_features_of)$^{[0,n]}$ OR (P130B.features_are_also_found_on)$^{[0,n]}$ }-> **C1.Object:**

{E24.Physical_Man-Made_Thing -- P62F.depicts **-> C1.Object:**

{ C1.Object --{(P46F.is_composed_of) $^{[0,n]}$ OR (P106F.is_composed_of) $^{[0,n]}$ OR (P148F.has_component) $^{[0,n]}$}-> **C1.Object** [-- P2F.has_type -> **E55.Type**]

}

OR

E89.Propositional_Object --P67F.refers_to-> **C1.Object:**

{ C1.Object --{(P46F.is_composed_of) [0,n] OR
(P106F.is_composed_of) [0,n] OR
(P148F.has_component) [0,n]}-> **C1.Object** [--
P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P128F.carries ->
E73.Information_Object:

{E73.Information_Object -- P67F.refers_to -> **C1.Object:**

{ C1.Object --{(P46F.is_composed_of) [0,n] OR
(P106F.is_composed_of) [0,n] OR
(P148F.has_component) [0,n]}-> **C1.Object** [--
P2F.has_type -> **E55.Type**]

}

}

}

}

}

}

b. is referred to by

An Event may be referred to by Things that have as theme or subject the Event, or that refer to or are about the Event. We may even expand the reference to Events that contain the Event in reference.

In CIDOC-CRM properties that define the "is referred to by" relationship are:

- P67B.is_referred_to_by
- P62B.is_depicted_by

An example from real metadata would be:

Event** is referred to by **Troy (the movie)

Trojan War -- P67B.is_referred_to_by ->**Troy**

The respective general fundamental relationship "*Event is referred to by Thing"* is:

E5.Event --(P9B.forms_part_of)$^{[0,n]}$ **-> E5.Event:**

{E5.Event--P67B.is_referred_to_by -> **E89.Propositional_Object:**

{**E89.Propositional_Object --** { (P130F.shows_features_of)$^{[0,n]}$ OR (P130B.features_are_also_found_on)$^{[0,n]}$ }-> **E89.Propositional_Object:**

{ E89.Propositional_Object --(P148F.has_component) $^{[0,n]}$-> **E89.Propositional_Object:**

{E89.Propositional_Object [--P2F.has_type -> **E55.Type**]

OR

E73.Information_Object -- P128B.is_carried_by -> **E24.Physical_Man-Made_Thing:**

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) $^{0,n)}$ -> **E24.Physical_Man-Made_Thing**[--P2F.has_type -> **E55.Type**]

}

}

}

}

OR

E5.Event -- P62B.is_depicted_by-> **E24.Physical_Man-Made_Thing**:

{ **E24.Physical_Man-Made_Thing --** { (P130F.shows_features_of)$^{[0,n]}$ OR (P130B.features_are_also_found_on)$^{[0,n]}$ }-> **E24.Physical_Man-Made_Thing:**

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) $^{0,n)}$ ->

E24.Physical_Man-Made_Thing[--P2F.has_type -> **E55.Type**]

}

}

}

c. has met

This is a generic relationship used to return all the Events that a certain Thing has met. In other words, this means that a Thing has been present at the Event.

An example from real metadata would be:

Event** has met **Guernica painting

Move of Guernica painting from El Prado Museum to Museum National Reina Sofia --

P12F.occurred_in_the_presence_of ->**Guernica Painting**

The respective general fundamental relationship "*Event has met Thing"* is:

E5.Event --(P9F.consists_of)$^{[0,n]}$ -> **E5.Event:**

{E5.Event-- P12F.occurred_in_the_presence_of -> **C1.Object:**

{C1.Object --{(P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n]
OR (P148F.has_component) [0,n]}-> **C1.Object** [--P2F.has_type -> **E55.Type**]

}

}

Specializations:

For the "has met" relationship we can define three sub-relationships, in order to be able to be more specific for actions performed on things during an event. This is useful because users are likely to be interested in certain things included in an Event such as the "created", "destroyed", "modified" and "used" Things. So we define:

- created

This sub-FR is used to relate the Thing with the Events of its creation. One may think that a Thing is created during one event and not many, but also the super events of the creation event may be considered as creation events.

An example from real metadata would be:

Event** created **Guernica painting

Guernica Painting Event -- P92F.brought_into_existence ->**Guernica Painting**

The respective general fundamental relationship "*Event created Thing"* is:

E5.Event --(P9F.consists_of)[0,n] **-> E5.Event*:***

{ E63.Beginning_of_Existence --P92F.brought_into_existence -> **C1.Object:**

{C1.Object --{(P46F.is_composed_of) [0,n] OR (P106F.is_composed_of) [0,n]
OR (P148F.has_component) [0,n]}-> **C1.Object** [--P2F.has_type -> **E55.Type**]

}

}

- destroyed

This sub-FR is used to relate the Thing with the Events of its destruction. One may think that a Thing is destroyed during one event and not many, but also the super events of the dstruction event may be considered as destruction events.

An example from real metadata would be:

Event** destroyed **Parthenon

Bombing of Acropolis Event -- P13F.destroyed ->**Parthenon**

The respective general fundamental relationship "*Event destroyed Thing"* is:

E5.Event --(P9F.consists_of)$^{[0,n]}$ -> **E5.Event:**

{ E64.End_of_Existence -- P93F.took_out_of_existence ->**C1.Object:**

{C1.Object --{(P46F.is_composed_of) $^{[0,n]}$ OR (P106F.is_composed_of) $^{[0,n]}$ OR (P148F.has_component) $^{[0,n]}$}-> **C1.Object** [--P2F.has_type -> **E55.Type**]

}

}

- modified

This sub-FR is used to relate the Thing with the Events of its modification

An example from real metadata would be:

Event** modified **Erechtheum

Erechtheum Modification by the Romans-- P31F.has_modified -> **Erechtheum**

The respective fundamental relationship "*Event modified Thing"* is:

E5.Event --(P9F.consists_of)$^{[0,n]}$ ->**E11.Modification**:

{**E11.Modification**-- P31F.has_modified->**E24.Physical_Man-Made_Thing**:

{**E24.Physical_Man-Made_Thing** --(P4F.is_composed_of)$^{[0,n]}$->

E18.Physical_Thing[--P2F.has_type -> **E55.Type**]

}

}

- used

This sub-FR is used to relate the Thing with Events in which it has been used.

An example from real metadata would be:

Event** used **Mylonas rifle

Thessaly Campaignof 1878 -- P16F.used_specific_object-> **greek rifle acq.No 32** --P2F.has_type -> **Mylonas rifle**

This is an example where either the user does not know or may not be interested in the exact name of the Thing used, but its type. So they want to know all the Events where Things of type *Mylonas rifle* were used.

The respective fundamental relationship "*Event used Thing"* is:

E5.Event --(P9F.consists_of)$^{[0,n]}$ ->**E7.Activity**:

{**E7.Activity** -- P16F.used_specific_object -> **C1.Object:**

{C1.Object --{(P46F.is_composed_of) $^{[0,n]}$ OR (P106F.is_composed_of) $^{[0,n]}$ OR (P148F.has_component) $^{[0,n]}$}-> **C1.Object** [--P2F.has_type -> **E55.Type**]

}

OR

E7.Activity -- P125F.used_object_of_type -> **E55.Type**

}

Event-Actor

a. refers to or is about

An event may refer to an Actor by Things that are created or produced in the Event and refer to or are about the Actor.

An example from real metadata would be:

Event *refers to or is about* ***Kazantzakis***

The writing of the original book " Nikos Kazantzakis" by Peter Bien -- P94F.has_created->

"Nikos Kazantzakis" original book -- P129F.is_about ->**Kazantzakis**

The respective general fundamental relationship "*Event refers to or is about Actor"* is:

E5.Event -- (P9F.consists_of)$^{[0,n]}$-> **E7.Activity:**

{E63.Beginning_of_Existence --P92F.brought_into_existence ->**C1.Object:**

{C1.Object --{(P46F.is_composed_of) $^{[0,n]}$ OR (P106F.is_composed_of) $^{[0,n]}$ OR (P148F.has_component) $^{[0,n]}$}-> **C1.Object:**

{**C1.Object--** { (P130F.shows_features_of)$^{[0,n]}$ OR (P130B.features_are_also_found_on) $^{[0,n]}$ }-> **C1.Object:**

{E24.Physical_Man-Made_Thing -- P62F.depicts **->**

E39.Actor:

{ E39.Actor --

(P107F.has_current_or_former_member_of)$^{[0,n]}$ ->

E39.Actor [--P2F.has_type -> **E55.Type**]

}

OR

E89.Propositional_Object --P67F.refers_to-> **E39.Actor:**

{ E39.Actor --

(P107F.has_current_or_former_member_of)$^{[0,n]}$ ->

E39.Actor [--P2F.has_type -> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P128F.carries ->
E73.Information_Object:

{E73.Information_Object --P67F.refers_to **E39.Actor:**

{ E39.Actor --

(P107F.has_current_or_former_member_of)$^{[0,n]}$ ->

E39.Actor [--P2F.has_type -> **E55.Type**]

}

}

}

}

}

}

b. is referred to by

An actor may refer to an Event by material or immaterial Things they create and which refer to some Event.

In CIDOC-CRM properties that define the "is referred to by" relationship are:

- P67B.is_referred_to_by
- P62B.is_depicted_by

An example from real metadata would be:

Event** is referred to by **Homer

The Trojan war --P129B.is_subject_of**->"Ilad"**-- P94B.was_created_by -> **"Iliad creation event"** -- P14F.carried_out_by -> **Homer**

OR

The return of Odysseus to Ithaca -- P67B.is_referred_to_by **->"Odyssey"**-- P94B.was_created_by -> **"Odyssey creation event"** -- P14F.carried_out_by -> **Homer**

The respective general fundamental relationship "*Event is referred to by Actor*" is:

E5.Event --(P9B.forms_part_of)$^{[0,n]}$ **-> E5.Event:**

{E5.Event --P67B.is_referred_to_by -> **E89.Propositional_Object:**

{**E89.Propositional_Object --** { (P130F.shows_features_of)$^{[0,n]}$ OR (P130B.features_are_also_found_on)$^{[0,n]}$ }-> **E89.Propositional_Object:**

{ E89.Propositional_Object --(P148F.has_component) $^{[0,n]}$-> **E89.Propositional_Object:**

{E89.Propositional_Object -- P94B.was_created_by -> **E65.Creation:**

{ E65.Creation -- (P9B.forms_part_of)$^{[0,n]}$ -> **E65.Creation**:

{ E65.Creation-- P14F.carried_out_by -> **E39.Actor:**

{ E39.Actor --
(P107B.is_current_or_former_member_of)
[0,n] -> **E39.Actor** [--P2F.has_type ->
E55.Type]

}

}

}

OR

E73.Information_Object -- P128B.is_carried_by -> **E24.Physical_Man-Made_Thing:**

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) 0,n) -> **E24.Physical_Man-Made_Thing**:

{ E24.Physical_Man-Made_Thing--
P108B.was_produced_by -> **E12.Production:**

{ E12.Production -- (9B.forms_part_of)[0,n] -> **E12.Production**:

{E12.Production -- P14F.carried_out_by-> **E39.Actor:**

{ E39.Actor --
(P107B.is_current_or_former_member_of) [0,n] -> **E39.Actor** [--
P2F.has_type -> **E55.Type**]

}

}

}

}

}

}

OR

E5.Event -- P62B.is_depicted_by-> **E24.Physical_Man-Made_Thing:**

{ **E24.Physical_Man-Made_Thing --** { (P130F.shows_features_of)[0,n] OR (P130B.features_are_also_found_on)[0,n] }-> **E24.Physical_Man-Made_Thing:**

{ E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) [0,n] -> **{ E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing--P108B.was_produced_by -> **E12.Production:**

{E12.Production -- (P9B.forms_part_of) [0,n] -> **E12.Production**:

{E12.Production -- P14F.carried_out_by-> **E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of) [0,n] -> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

}

}

}

}

}

c. by

Events are usually carried out by Actors, and by this FR one can query for all the Events carried out by certain Actors. Also the *by* FR includes the influenced by relation between the two Categories.

An example from real metadata would be:

Event** by **Alexander the Great

Conquest of the Persian Empire -- P14F.carried_out_by **-> Alexander the Great**

The respective general fundamental relationship "*Event by Actor*" is:

E5.Event-- (P9B.forms_part_of)$^{[0,n]}$-> **E7.Activity:**

{E7.Activity--P14F.carried_out_by->**E39.Actor:**

{E39.Actor--(P107B.is_current_or_former_member_of)$^{[0,n]}$-> **E39.Actor:**

{ E39.Actor --(P107F.has_current_or_former_member_of)$^{[0,n]}$ ->
E39.Actor [--P2F.has_type -> **E55.Type**]

}

}

OR

E7.Activity -- P15F.was_influenced_by-> **E39.Actor:**

{E39.Actor--(P107B.is_current_or_former_member_of)$^{[0,n]}$->**E39.Actor** [--
P2F.has_type -> **E55.Type**]

}

}

d. has met

This is a general relationship that connects an Event with actors that were present at the Event.

An example from real metadata would be:

Event** has met **Alexander the Great

Conquest of the Persian Empire -- P9F.consists_of **-> Conquest of Babylonia** -- P12F.occurred_in_the_presence_of **-> Alexander the Great**

The respective general fundamental relationship "*Event has met Actor*" is:

E5.Event--(P9F.consists_of)$^{[0,n]}$ -> **E5.Event:**

{**E5.Event** -- P12F.occurred_in_the_presence_of-> **E39.Actor:**

{E39.Actor--(P107B.is_current_or_former_member_of)$^{[0,n]}$->**E39.Actor** [-- P2F.has_type -> **E55.Type**]

}

}

Specializations:

In the fundamental relationship *has met* we can also define some sub-fundamental relationships, which contain more restricted information. This information is however commonly asked for by the users and a more specialized query may return better results based to what they actually want to know about.

- Brought into existence

With this specification we query only for events that brought into existence the specific Actor.

An example from real metadata would be:

***Event** brought into existence Philip II of Macedon*

Birth of Philip II of Macedon -- P92F.brought_into_existence ->Philip II of Macedon

The respective general fundamental relationship "*Event brought into existence Actor"* is:

E5.Event--(P9F.consists_of)$^{[0,n]}$ -> **E5.Event:**

{E63.Beginning_of_Existence --P92F.brought_into_existence-> **E39.Actor:**

{ E39.Actor --(P107F.has_current_or_former_member_of)$^{[0,n]}$ -> **E39.Actor** [--P2F.has_type -> **E55.Type**]

}

}

- Took out of existence

With this specification we query only for events that took out of existence the specific Actor.

An example from real metadata would be:

***Event** took out of existence Philip II of Macedon*

Murder of Philip II of Macedon -- P93F.took_out_of_existence ->Philip II of Macedon

The respective fundamental relationship "*Event took out of existence Actor"* is:

E5.Event--(P9F.consists_of)$^{[0,n]}$ -> **E5.Event:**

{E64.End_of_Existence -- P93F.took_out_of_existence -> **E39.Actor:**

{ E39.Actor --(P107F.has_current_or_former_member_of)[0,n] -> **E39.Actor** [

--P2F.has_type -> **E55.Type**]

}

}

Event-Event

The Event-Event fundamental relationships can be switched to Event-Time fundamental relationships, by further adding the CIDOC-CRM property P4F.has_time-span at the range category Event **(E5.Event** --P4F.has_time-span->**E52.Time-Span)**. This happens because Time refers to the chronological definition of Events.

a. refers to or is about

An event may refer to an Event by Things that are created or produced in the Event and refer to or are about the Event.

An example from real metadata would be:

Event** refers to or is about **Olympic Games of 1896

The writing of the book "Olympic Games in Athens 1896-1906" by Kardasis --

P94F.has_created-> **"Olympic Games in Athens 1896-1906" book --** P67F.refers_to ->

Olympic Games of 1896

The respective general fundamental relationship "*Event refers to or is about Event"* is:

E5.Event -- (P9F.consists_of)$^{[0,n]}$-> **E5.Event:**

{E63.Beginning_of_Existence --P92F.brought_into_existence ->**C1.Object:**

{C1.Object --{(P46F.is_composed_of) $^{[0,n]}$ OR (P106F.is_composed_of) $^{[0,n]}$ OR (P148F.has_component) $^{[0,n]}$}-> **C1.Object:**

{**C1.Object--** { (P130F.shows_features_of)$^{[0,n]}$ OR (P130B.features_are_also_found_on) $^{[0,n]}$ }-> **C1.Object:**

{E24.Physical_Man-Made_Thing -- P62F.depicts **-> E5.Event:**

{**E5.Event** -- (P9F.consists_of)$^{[0,n]}$->**E5.Event** [-- P2F.has_type-> **E55.Type**]

}

OR

E89.Propositional_Object -- P67F.refers_to-> **E5.Event:**

{**E5.Event** -- (P9F.consists_of)$^{[0,n]}$->**E5.Event** [-- P2F.has_type-> **E55.Type**]

}

OR

E24.Physical_Man-Made_Thing -- P128F.carries -> **E73.Information_Object:**

{E73.Information_Object -- P67F.refers_to ->

E5.Event:

{**E5.Event** -- (P9F.consists_of)$^{[0,n]}$->**E5.Event** [-- P2F.has_type-> **E55.Type**]

}

}

}

}

}

}

b. is referred to at

An Event may be referred by Events during which Things that refer to the Event are created.

In CIDOC-CRM properties that define the "is referred to at" relationship are:

- P67B.is_referred_to_by
- P62B.is_depicted_by

An example from real metadata would be:

Event** is referred to at "**Leonidas at Thermopylae" Painting Event

Greco-Persian War-- P9F.consists_of-> **Thermopylae Battle** -> P67B.is_referred_to_by -> "**Leonidas at Thermopylae" painting --** P94B.was_created_by -> **"Leonidas at Thermopylae" Painting Event**

The respective general fundamental relationship "*Event is referred to at Event"* is:

E5.Event -- (P9F.consists_of)$^{[0,n]}$->**E5.Event:**

{E5.Event--P67B.is_referred_to_by -> **E89.Propositional_Object:**

{**E89.Propositional_Object --** { (P130F.shows_features_of)$^{[0,n]}$ OR (P130B.features_are_also_found_on)$^{[0,n]}$ }-> **E89.Propositional_Object:**

{ E89.Propositional_Object --(P148F.has_component) [0,n]-> **E89.Propositional_Object:**

{E89.Propositional_Object -- P94B.was_created_by -> **E65.Creation:**

{ E65.Creation -- (P9B.forms_part_of)[0,n] -> **E5.Event** [--P2F.has_type -> **E55.Type**]

}

OR

E73.Information_Object -- P128B.is_carried_by -> **E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) 0,n) -> **E24.Physical_Man-Made_Thing**:

{E24.Physical_Man-Made_Thing-- P108B.was_produced_by -> **E12.Production:**

{ E12.Production -- (P9B.forms_part_of)[0,n] -> **E5.Event** [-- P2F.has_type -> **E55.Type**]

}

}

}

}

}

OR

E5.Event-- P62B.is_depicted_by-> **E24.Physical_Man-Made_Thing:**

{ **E24.Physical_Man-Made_Thing --** { (P130F.shows_features_of)[0,n] OR (P130B.features_are_also_found_on)[0,n] }-> **E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing -- (P46B.forms_part_of) [0,n] -> **{ E24.Physical_Man-Made_Thing:**

{E24.Physical_Man-Made_Thing--P108B.was_produced_by -> **E12.Production:**

{ E12.Production -- (P9B.forms_part_of)[0,n] -> **E5.Event** [--P2F.has_type -> **E55.Type**]

}

}

}

}

}

c. from

Events may occur in the context of a bigger event or may occur concurrently with another event.

An example from real metadata would be:

Event** from **World War II

Greco-Italian War -- P10F.falls_within -> **World War II**

OR

The Battle of Crete -- P10F.falls_within -> **The German Occupation** -- P10F.falls_within -> **World War II**

The respective general fundamental relationship "*Event from Event*" is:

E5.Event--{ (P9B.forms_part_of)[0,n] OR (P119F.meets_in_time_with) [0,n] OR (P119B.is_met_in_time_by) [0,n] OR (P118F.overlaps_in_time_with) [0,n] OR (P118B.is_overlapped_in_time_by) [0,n] OR (P117F.occurs_during) [0,n] OR (P114F.is_equal_in_time_to) [0,n] OR (P10F.falls_within) [0,n] }-> **E5.Event** [--P2F.has_type -> **E55.Type**]

d. has part

Events may contain other events or may co-occur with other events. It is the reverse of the *Event from Event* FR, although the relations that are about the co-occurrence of the two Events.

An example from real metadata would be:

Event** has part **The Battle of Crete

The German Invasion -- P119F.meets_in_time_with -> **The battle of Crete**

OR

The Battle of Crete -- P10B.contains -> **The German Occupation** -- P10B.contains -> **The battle of Crete**

The respective general fundamental relationship "*Event has part Event*" is:

E5.Event--{(P9F.consists_of)[0,n]OR(P119F.meets_in_time_with) [0,n] OR (P119B.is_met_in_time_by) [0,n] (P118F.overlaps_in_time_with) [0,n] OR (P118B.is_overlapped_in_time_by) [0,n] OR(P117B.includes)[0,n] OR (P114F.is_equal_in_time_to) (0,n)OR(P10B.contains) [0,n]} -> **E5.Event**[--P2F.has_type -> **E55.Type**]

Event-Concept

a. has type

This relationship connects an Event with its type, which describes its concept. An Event may belong to more than one type categories.

An example from real metadata would be:

Event** has type **War

World War II -- P2F.has_type->**War**

The respective fundamental relationship "*Event has type Concept"* is:

E5.Event--(P9F.consists_of)[0,n]-> **E5.Event:**

{E5.Event-- P2F.has_type-> **E55.Type**:

{ **E55.Type** -- {(P127F.has_broader_term)[0,n] OR (P2B.is_type_of)[0,n]}-> **E55.Type** [P2F.has_type -> **E55.Type**]

}

}

CONCEPT

Concept is the Fundamental Category comprising the Type(s) that instances of the other Fundamental Categories have. This is a separate category from Thing concerning the discourse of cultural heritage and more specifically the digitization of cultural heritage objects. Instances of Concept are universal as they categorize particular instances whereas instances of Things are particular since they have specific identification and properties. Here the declaration of Concept does is not specified as it is not concerned to be a Thing. In other information systems that describe Types, such as in Biology, Types could be concerned as Things.

To link instances of Concept to instances of other FCs we use the *has type* property.

Concept-Place

a. is type of

Concepts that comprise the Types of a certain Place, including the types of its sub-parts.

An example from real metadata would be:

Concept** is type of **Greece

Country-- P2B.is_type_of ->**Greece**

The respective fundamental relationship "*Concept is type of Place"* is:

E55.Type -- {(P127B.has_narrower_term)$^{[0,n]}$ OR (P2B.is_type_of)$^{[0,n]}$}-> **E55.Type:**
{E55.Type --P2B.is_type_of->**E53.Place:**
{E53.Place -- (P89F.falls_within)$^{[0,n]}$-> **E53.Place**
}
}

Concept-Thing

a. is type of

Concepts that comprise the Types of a certain Thing.

An example from real metadata would be:

Concept** is type of **Guernica Painting

Surrealist-- P2B.is_type_of ->**Guernica Painting**

The respective fundamental relationship "*Concept is type of Thing"* is:

E55.Type -- {(P127B.has_narrower_term)$^{[0,n]}$ OR (P2B.is_type_of)$^{[0,n]}$}-> **E55.Type:**

{E55.Type --P2B.is_type_of->**C1.Object:**

{C1.Object -- { (P46B.forms_part_of) $^{[0,n]}$ OR (P106B.forms_part_of) $^{[0,n]}$ OR (P148B.is_component_of) $^{[0,n]}$ } -> **C1.Object**

}

}

Concept-Actor

a. is type of

Concepts that comprise the Types of a certain Thing.

An example from real metadata would be:

Concept** is type of **Picasso

Painter-- P2B.is_type_of ->**Picasso**

The respective fundamental relationship "*Concept is type of Actor"* is:

E55.Type -- {(P127B.has_narrower_term)$^{[0,n]}$ OR (P2B.is_type_of)$^{[0,n]}$}-> **E55.Type:**
{E55.Type --P2B.is_type_of->**E39.Actor:**

{E39.Actor -- (P107B.is_current_or_former_member_of)$^{[0,n]}$-> **E39.Actor**

}

}

Concept-Event

a. is type of

Concepts that comprise the Types of a certain Event.

An example from real metadata would be:

Concept** is type of **Excavation Event at Knossos

Excavation-- P2B.is_type_of -> **Excavation Event at Knossos**

The respective fundamental relationship "*Concept is type of Event"* is:

E55.Type -- {(P127B.has_narrower_term)$^{[0,n]}$ OR (P2B.is_type_of)$^{[0,n]}$}-> **E55.Type:**
{E55.Type --P2B.is_type_of->**E5.Event:**

{ E5.Event -- (P9B.forms_part_of)$^{[0,n]}$-> **E5.Event**

}

}

Concept-Concept

a. has type

A Concept may have broader terms with which it is expressed, and with this FR we can get all the broader terms of this type.

An example from real metadata would be:

Concept** has type **painting

Oil Painting-- P127F.has_broader_term -> **painting**

The respective fundamental relationship "*Concept has type Concept"* is:

E55.Type--{(P127F.has_broader_term)$^{[0,n]}$ OR (P2F.has_type)$^{[0,n]}$}-> **E55.Type**

b. is type of

A Concept may have narrower terms with which it is expressed, and with this FR we can get all the narrower terms of this type.

An example from real metadata would be:

Concept** is type of **Armchair

Furniture-- P127B.has_narrower_term -> **Armchair**

The respective fundamental relationship "*Concept is type of Concept"* is:

E55.Type--{(P127B.has_narrower_term)$^{[0,n]}$ OR (P2B.is_type_of)$^{[0,n]}$}-> **E55.Type**