**Parthenos Project -**    
ICS-FORTH (Foundation for Research and Technology – Hellas)  
Institute of Computer Science  
N. Plastira 100  
Vassilika Vouton, GR-700 13 Heraklion, Crete, Greece

**Summer**

17

08

**Fall**

Principles for Modelling Ontologies: A Short Reference Guide

Parthenos / ICS-FORTH

This document is a first draft of an ontological modelling reference booklet designed to aid modellers developing or extending ontologies intended to support data integration for an empirical research domain

Editor: George Bruseker

Contributors: Francesco Beretta, Maria Daskalaki, Emiliano Degl’Innocenti, Martin Doerr, Øvyind Eide, Achille Felicetti, Anaïs Guillem, Dimitris Kafetzopoulos, Fahad Khan, Carlo Meghini, Nicola Spyratos, Maria Theodoridou, Athanasios Velios

Date of Creation: July 2017

Last Update: November 2022

Version 0.2

Acknowledgements: Parthenos, FORTH + more

**Table of Contents**

[**General Introduction**](#_nk4zxwbnppug)4

[Theoretical Background](#_27oh55vf3a33) 5

[**Process Model**](#_ne4q5se3pgea)10

[Phase A: Purpose Definition](#_lchdal8qwj8y) 11

[Phase B: Ontology Constructs Definition](#_xiyfmxpz7iel) 12

[Phase C: Implementation and publishing](#_kbtn6q57cnyj) 14

[Mapping](#_fmtqya6nyde) 14

[**Principles Introduction**](#_30j0zll)18

[**Glossary**](#_3fv7ayo91pjc)20

[**Engineering from an Empirical Base**](#_1fob9te)25

[1.1 Model from existing / actually used structured information sources (whenever available)](#_2et92p0) 26

[1.2 Model according to the research questions justifying the structured information](#_tyjcwt) 27

[1.3 Model from actual information values](#_3dy6vkm) 28

[**Knowledge Structure Basics**](#_1t3h5sf)29

[2.1 Detect hidden relations in terms](#_2s8eyo1) 30

[2.2 Distinguish particulars from universals in the target domain](#_3rdcrjn) 31

[2.3 Do not define the same property twice for different classes. Find the superclass for it](#_26in1rg) 32

[2.4 IsA is an increase of instances and a decrease of properties](#_lnxbz9) 32

[**Concept Relevance**](#_r6wxve9igk3t)35

[3.1 Model primitive concepts first](#_1ksv4uv) 36

[3.2 A class should allow the formulation of a query that answers a relevant question](#_44sinio) 37

[3.3 Model manageable units](#_z337ya) 38

[3.4 Model concepts that express the least interpretational position in order to make the model robust against revision](#_1y810tw) 39

[**Open World**](#_4i7ojhp)39

[4.1 Never define a class as complement](#_2xcytpi) 40

[4.2 Cover incomplete details of knowledge by what you do know](#_1ci93xb) 42

[4.3 Do not create closed worlds of properties](#_3whwml4) 43

[**Open World and Knowledge Progress**](#_2bn6wsx)44

[5.1 Support progressive improvement of classification knowledge by IsA hierarchy](#_qsh70q) 45

[5.2 Do not model conclusions before and without their reasons](#_3as4poj) 46

[5.3 Describe the intension of and declare classes that model the parts of the domain you understand](#_1pxezwc) 47

[5.4 Model domains and range or properties consistent with your level of knowledge of the domain of discourse](#_49x2ik5) 48

[**Open World and Knowledge Base**](#_2p2csry)49

[6.1 The absence of a property in the knowledge base is not its negation in reality](#_147n2zr) 50

[6.2 Allow alternatives or contradictions in the data](#_3o7alnk) 51

[6.3 Make sure alternative assertions can be unambiguously related to a single entity](#_23ckvvd) 52

[6.4 Explain Data Structures](#_ihv636) 53

[**Objectivity**](#_32hioqz)54

[7.1 Be view neutral](#_41mghml) 55

[7.2 Avoid concepts depending on a personal/ spectator perspective](#_2grqrue) 56

[7.3 Avoid concepts depending on accidental and uncontextual properties](#_vx1227) 57

[7.4 Maintain independence from scale](#_3fwokq0) 58

[**Language and Concepts**](#_1v1yuxt)58

[8.1 Don’t confuse polysemy with multiple abstractions](#_4f1mdlm) 60

[8.2 Most binary relationships in intuitive conceptualizations conceal temporal entities](#_2u6wntf) 61

[**Conceptual Modelling Checklist**](#_19c6y18)62

# 

# General Introduction

This document is a reference guide to the ontology engineering method underlying the construction of the CIDOC CRM (ISO21127:2014) and its extensions. It is intended as an accompanying handbook for users that have received respective training by courses or other didactic material and are in a process to use the CIDOC CRM or its extensions, to map legacy data structures to it, or to develop further extensions.

This guide describes a methodology for formal ontology development presented in the form of a simple process model of recommended steps and a series of distinct principles for consultation during conceptual modelling activities. These principles are deemed pertinent to the creation of information systems about scientific and scholarly knowledge using formal ontology, although in practice the scope of their application may be broader. The principles are arranged in thematic groups representing clusters of relevant, practical modelling issues and responses. They constitute a list of functional conclusions regarding key questions/problems recurrently encountered in conceptual modelling activities and which have been arrived at through over twenty-five years of hands on, empirical engagement in ontology construction.

## Ontologies, data models and information systems

In the software industry, a data model is usually a formalized description of how to organize data in an information system. In the development of data models the developers will focus on how to organize the data to fulfil the customers’ specification of the functionality of the system. The focus will be on data consistency, ease of maintenance, use of storage and performance. A good information system should also be in line with the users’ mental model of the task to be solved. The developers must try to understand this mental model and use it at least as a starting point in the development of the technical data model used in the implementation of the system. Ideally the technical data model should be a formalization of the mental model expressed in some software specification language, for example UML.[[1]](#footnote-1)

The border between technical data models and formalized conceptualization of parts of the real world is not clean-cut. There is a long tradition in computer science of using terms from psychology, linguistics and philosophy metaphorically. This adds to this fuzziness and blurs the distinctions between the real world, models of the real world and implementations of these models. This is especially visible in artificial intelligence, knowledge engineering and semantic technologies, the most innovative but also the most speculative fields of computer science.

Formal ontologies are a means to describe information structures. They are developed by computer scientists in collaboration with philosophers in order to overcome the compatibility issues present a result of idiosyncratic and mutually incompatible data models having been developed for the same subject matter by independent teams for the same domain. In order to develop a solution to this problem, the idea was and is to specify the relevant kinds of things (objects, people, places, events, time, concrete ideas etc.) and their possible relationships in the part of the universe covered by one or more information systems with the use of logical expressions.

This specification aims at representing exclusively the relevant possible states of affairs as currently commonly understood by domain experts, creators, and users of information systems. It must not compromise the correctness of description of the universe of discourse (“ontological commitment”) to implementation considerations described above.

The consequence of this necessary restriction is that data entry forms and their labels, system internal methods of storing and implementation, and query formulation tools and their terms presented to users will and must in general deviate from the formal ontology. This is the difference between the specification level and implementation level. To be sure that the implementation is correct with respect to the specification level, that is the ontology, the “deviations” must be able to be described in terms of logical relations of the formal ontology to the constructs in the implemented components. The use of a common formal ontology enforces a good practice of information modelling and supports the integration of different systems of overlapping domains and the migration of data to new and more powerful platforms.

## Theoretical Background

This introduction to the theoretical background outlines how the ground positions on formal ontology taken here differ from some other major schools of formal ontology. The differentiation arises out of the practical effort to apply existing methodologies and finding the need to take new ground positions about the task of ontology development itself and the limits that it faces precisely in order to meet the end of providing functional models for the fields under investigation.

The basic issue is to clarify what the product of a modelling effort is: what is a formal ontology? There is general agreement between different schools of modelling that a formal ontology is a tool for communication in distributed information environments. We, further, take the position that in scientific and scholarly communication, a model is necessarily a means of representation of some aspects of reality.

Where there is then room for disagreement relates to the issues of:

1. What kind of representational tool a formal ontology is.
2. What is the precision of the relation between a formal ontology and the reality it attempts to represent.
3. What aspect of reality a formal ontology attempts to represent and how this affects its verifiability.

Formal ontologies belong to the class of logical theories and, as such, belong to the general class of discrete mathematical theories. They are further distinguished by explicitly aiming to approximate reality and, therefore, are conditioned by their representative functionality – i.e. their connection to empirical reality - with regards to their truth functionality. The kinds of mathematical representation of reality include but are not limited to such different forms as continuous functions, neural networks, statistical models, and logical theories.

Scientists and many scholars make and use mathematical models of reality in order to communicate about it or to predict its behavior. The suitability and application of these models varies substantially depending on (i) the nature of the aspect of reality to be modelled, (ii) the cognitive capacity of some group to perceive and represent the reality intended by the model, and (iii) the features of the aspect of reality modelled which are of interest for scientific prediction/communication.

Thus far, the position presented in relation to issue (a) above should be largely in accord with the approaches of other schools of formal ontology. Where our position begins to differ lies in the conclusion we draw from this categorization based on our understanding of the limits of mathematical models as representational tools.

The representational success of any mathematical model referring to reality presupposes that some knowing agents (scientist, scholars and other users) are able to understand and then instantiate said model with relevant parameters taken from reality and, thereafter, can compare results of data represented in the model with actual situations in reality, and/or amongst themselves. The observational checking can as such be carried out through the natural senses as aided by artificial sensors.

Crucially, the above entails that the precision of applying mathematical models to reality is ultimately externally limited by the precision of the requisite parameter provision (data collection) and result comparison by knowing agents. This accuracy is necessarily empirically affected by the factors of (x) determinacy of the defined parameters in the reality itself[[2]](#footnote-2), (y) precision of the tool used to observe it, and (z) the precision of the interpretive frame operative to give sense to the results/data.

Given the above, a different position is taken here on issue (b) than schools that would argue that a well-built ontology should ideally hold a one-to-one relation between its concepts and the world it aims to represent. Instead, if we accept the representational mode of the ontology as a logical theory, this introduces necessarily a limit to the things it can represent at all as well as the potential precision of its representation.

First, a formal ontology is not suitable for representing all aspects of reality that can be named. Because of its representational form, it can only represent clearly identifiable individual items, out there in reality, with relations that can – in principle - be verified or falsified by independent persons. Examples of such items would be cars, hammers, a shoemaker, or a village, but would not include other aspects of reality such as “a” cloud, wave, or wind, all of which are better represented by other models suitable to representation of continuous phenomena.

Second, the representational precision of an ontology is limited by its form itself. Classes and relations specified at the level of the universal are always approximations. They cannot be expected to form an exact duplicate of the reality modelled. The inherent inability of an ontology to provide a one-to-one ideal surrogate for real world objects in no way, however, makes it useless. Rather, the ontology development process must start with identifying whether the degree of deviation of reality from each element of the model is tolerable for the intended communication purpose of the target community both in terms of precision and in terms of statistical occurrences of exceptions. If this is established, then logical rules for reasoning can be formulated and used as an additional engineering construct.

That being said, the lack of a one-to-one relation between a model and reality does not entail a subjectivist position on issue (c). Such a position is staked out by some schools of conceptual modelling[[3]](#footnote-3) that limit the goal of modelling of a formal ontology to replacing intuitive conceptualizations held by groups of actors by logical terms, arguing that how such representations then relate to the reality as such constitutes a black box outside the scope of modelling activity. That is to say that for those schools of thought, delving into the questions of cognitive science and how correspondence is established between the knower and some known thing is deliberately left outside of the questions posed by the conceptual modeller. This position is taken out of a principle of caution and as a practical measure to limit the scope of formal ontology to a reasonable scale, unburdening it from a direct scientific activity it is not responsible for.

However, the self-restriction that comes from taking this position is unnecessary and, more importantly, unhelpful to good modelling. Reference to reality is both a necessary and grounding factor for good modelling and leaving it out of scope leaves models without an objective referent. Such a limitation is also not justified from a pragmatic, epistemic perspective. There exist many things in our reality that people can reliably identify, classify, relate, and communicate in a form compatible with formal ontologies without using logical definitions. It is a fact that we survive in complex environments and that this is the legal basis for the normal liability of people for their actions in a complex social environment (despite exceptions, border cases and limited precision of expression). Therefore, reference to reality is functional in an engineering sense.

Such reference to reality is of course a function of highly complex factors, studied by cognitive sciences, of how communicative agents adjust to contexts of reference and subconsciously modify concepts in a dynamic and adaptive way and yet are still able to communicate. This flexibility and sophistication is something that formal ontologies and information technology do not have. The context of reference consists of a combination of the phenomena under consideration (for instance being in a clinic or on the highway) and the questions a communication tries to answer. It is the richness of knowledge behind “surface concepts” and the flexibility to adapt concepts to a changing reality, and not the deficiencies of knowing agents to formulate logically cogent conceptualizations, that hinders the immediate transfer of human concepts into an information environment.

For instance, most people are aware of the details of how a human being comes into existence. However, depending on the question, people will choose to regard it as having begun with conception or at birth. They may regard cut hair as a body part or as waste. This is not due to subjectivity or different perceptions of reality. They represent, rather, a function of selecting a suitable simplification of reality to communicate answers to queries such as “when can a human being become heir of someone” or “when should human life be protected”. Even within the same “domain” such multiple definitions may occur.

Consequently, ontology engineering methodology described here consists of:

1. Fixing a context of discourse with a limited subject matter and limited generic questions in order to reduce the interpretative complexity of concepts, a sort of “requirements specification”.
2. Exploring which concepts in this context can be formulated as a logical theory that can sufficiently represent the reality under consideration and support the intended generic questions.
3. Selecting or inventing concepts from the range of experience of experts, from documentation, and by actively learning about phenomena in the aspect of reality under investigation.

Since human experience is not ready-to-hand, but partially subconscious, and individual and collective knowledge is limited, this process of “knowledge engineering” consists of making participating experts consciously aware of the relevant aspects of reality, and facilitates them to learn from each other, from documentation and even active observation about exceptions, border cases etc.

Since this is a process of systematically increasing collective knowledge, it implies that the basis for a formal ontology (that has the required quality of representing reality) is a limited set of knowledge, which is expected to grow, but never be exhaustive. Therefore, the methodology contains a set of innovative advice on how to foresee the effect of new facts and to evolve the ontology from “safe grounds of knowledge”. The “Open World Assumption” is such a principle known from computer science, which is generalized here for that purpose.

For the above reasons, this guide proposes a process we found apt to help elicit systematically the relevant knowledge and to generalize it and widen its scope. It also proposes a set of principles to be applied alongside this process that aim to facilitate the work of a team of conceptual modellers whose goal is to build a functional formal ontology useful for some empirical domain and group of users.

# The Process Model

The following is a compact description of the CRM ontology engineering process, as it has emerged from over twenty-five years of practice. This process is similar but substantially different from processes that have been described going back to Booch and Coud-Jordan for conceptual modelling with object-oriented programming languages, and which often have uncritically been reapplied to ontology engineering. The major differences are (i) that we identify classes for the relevant properties instead of properties for relevant classes, and (ii) that the process is completely iterative. Another substantial difference is that we deal only with ontologies made to formulate propositions in information systems, as originally described by Thomas Gruber [XXX]. We do not apply this methodology to create terminological systems (vocabularies, thesauri, classification systems, typologies, etc.) to be used as data in propositions in the target systems, such as the Getty Art & Architecture Thesaurus (AAT) or Library of Congress Subject Headings (LCSH)

In the process of ontology engineering under the above restriction we can distinguish two different starting situations:

1. Building an ontology “from scratch” for a new domain.
2. Building an ontology from a functional set of information structures in use for a domain.

In both cases we assume (i) that the engineer is aware of the CIDOC CRM as a set of concepts for reuse in the process we will describe below, and (ii) that the engineer will recognize if a concept emerging in the process has already been described more or less in the CRMbase, an extension of it, or in another ontology. However, this must not create a bias to “press” things under CRM concepts, but rather be each time a test of the adequacy of the CRM. Some concepts in the CRM will most likely be generic enough to be adequate for any ontology dealing with the past. For some other concepts, a slight expansion of its definition can be functional, as long as that does not create other inconsistency in the CRM. Otherwise, new concepts should be created without any bias with regards to the existing ones. Therefore, we describe the process initially in a way as if no prior ontology would exist, followed by describing the process of actually “mapping” to the CRM, i.e., fitting concepts under the CRM, as a special aspect of the general engineering process.

**The Process**

This process is iterative. Any step may be occasion to reconsider previous ones. Each iteration improves the overall understanding. Care must be taken to recognize and break circular arguments.

The process consists of 3 phases:

1. Definition of purpose.
2. Ontology construction.
3. Implementation and publishing.

## Phase A: Purpose Definition

The steps are:

* 1. **Define the theoretical scope**, e.g., “political history”, “recording art conservation”, “history of music and performances”, etc. This restricts the kind of phenomena under consideration to things studied or to be studied by one or more explicit disciplines.
  2. **Define the overall question** to an information system, for instance “impact of technology on material conditions of living”, or “evidence for archaeological opinions documented in the excavation process”. It is good practice to explicitly exclude topics in terms of overly details or rareness. Each relation to be defined in the next phase can be regarded as element in a formalized question, a so-called query, which should contribute to the overall question according to the experts’ opinion.
  3. **Define the empirical source material** (“practical scope”) to be used to elicit the relevant concepts. These can include texts, images, videos, datasets, and information structures. Information structures must be accompanied with representative datasets using them.
     + Case (1): the scope of the target ontology is **limited** to what is relevant in and meant by these sources. E.g., FRBRoo is limited to what is directly or indirectly meant by FRBR documents – texts defining data structures for future use.
     + Case (2): The sources are a **representative** selection with sufficient coverage and may later be extended. E.g., CRMbase was initially engineered from museum collection management systems.
     + In case (a), building an ontology from scratch, it is advisable that a team of experts extracts intuitive entities and relations from a reasonable set of sources, typically texts, and uses them as surrogate for existing data structures that relate to the overall questions.
     + In case (b), building an ontology from a particular set of used information structures, their successful use is taken as evidence for the relevance of the underlying concepts. E.g., CRMbase is limited to senses directly or indirectly meant by existing and used information structures (we will call the elements of preexisting information systems also “intuitive terms” without implying a prejudice).

## Phase B: Ontology Constructs Definition

The steps are:

1. **Take a list of intuitive, specific terms** from the empirical source materials. These can be relations, entities from information structures, or categorical terms in the narrower sense. Even though looking for properties and relations is reserved for the next step, it is difficult to imagine relations without conceptualizing the entities they relate. However, it is advisable to avoid making abstract generalizations in this step because the properties are best understood in very specific cases. In information structures, strings and Boolean values may hide concepts!
2. **Create a list of properties for the intuitive entities.**  Find their relevant properties (behavior) for the discourse about the intended overall questions. Discuss all known reality within the scope. Each property must answer a specific question that helps understanding the overall questions. Detect and resolve polysemy by splitting terms into multiple concepts if necessary (“Where was the university when it decided to take more students?”). Change mentally the context of use of a term and observe its properties (e.g., is “pencil” a name of an object in a museum and in a shop?).
3. **Detect entities hidden in intuitive relations**. These are most frequently events, activities that are either shortcut by a (binary) relation or that initiate or terminate a relation. This may reveal that some properties from step 2 are not relevant, in which case, revise step 2. There may be source material supporting the hidden entities, in which case, revise step 1 and possibly Phase A. Truly n-ary relations are relatively rare (e.g., relative positions), but if encountered they must also be modelled as entities in the current knowledge representation languages.
4. **Detect classes from properties.** In this step, use only properties that are not logical deductions from other properties (e.g. « creator of » is a deduction from « created by – activity – carried out by »). Find the general classes for which each property is characteristic. In other terms, find the one most specific class that generalizes over all classes for which you are sure that the property applies as domain or range.
5. **Provide identity conditions to the classes.** Answer the questions:
   * By what something be determined as instance of this class?
   * Is there something that is and that definitely is not an instance of this class?
   * What makes an instance distinct from another and be the same after some time?
   * What belongs to it as extent or part? How do instances come into being/ end being?

Discuss all known reality within the scope. Seek expert knowledge about exceptions. Find out if this concept has been thought of or been defined somewhere. Try to learn or adopt better definitions.

1. **Create class hierarchies.** Determine which of the new classes are superclasses of others, i.e., if a domain (range) of a property A as defined above generalizes (possibly besides others) over a domain (range) of a property B. In other words, wherever B applies, A applies, but not vice-versa. This process may raise questions about the initial properties, in which case, revise starting at step 2.
2. **Create property hierarchies.** Find out, which properties imply other properties. E.g., in order to carry out an activity, one must be present at the activity. This process may raise questions about the initial properties and classes, in which case, revise starting at step 2.
3. **Conduct a property consistency test.** This is the ultimate test of adequacy. Test, which and if combinations properties complement each other to answer more complex relevant questions. Test if the specificity of domains and ranges of complementing properties are compatible. (For instance, can some kinds of things have a dimension, but cannot be measured?– if yes, which is the process to determine it?). Find gaps in the reasoning, find modelling patterns and check if variants indicate gaps. According to the application, purely logically derived concepts may be added. Test that **logical consistency** applies throughout the model. If issues are identified during testing, revise starting at step 2.
4. **Reduce the model.** Remove properties and classes not needed to implement the required functions. Keep them aside for possible future extensions. The smaller the model, the more effective is the information system.

## 

## Phase C: Implementation and publishing:

The steps are:

1. **Implement the model** in a specific knowledge representation language with a specific syntax. For instance, XML RDF/OWL, TRIG, etc. Verify logical consistency by S/W. According to the application, other purely logically derived concepts may be added.
2. **Write a textual definition** with a clear introduction reporting Phase A, all constructs in a syntax independent format that clearly relates to the logic of the model. This may be enriched with FOL statements (Second Order Logic?). Write extensive scope notes to clarify all identity conditions not put in terms of logic.
3. **Install curation.** Install a maintenance team. Identify the authority and authoritative procedure for updates (this can be completely democratic). Provide a public site with a transparent release management and make copies of the implementation and text available under transparent conditions.

## Mapping

By “mapping” we mean the translation of machine-readable instructions that an automated algorithm can follow in order to transform a set of data organized following a schema A (“source schema”) into a set of data organized following a schema B (“target schema”) ideally without loss or change of meaning. This procedure is used either for migrating data from one format and/or database to another, or for “providing” a copy of some “source” data to a “target” data aggregation service.

In practice, the ideal complete correspondence must be modified by distinguishing the following situations:

1. Only a part of the source schema is of interest for the intended transfer
2. The target schema is more general but less precise than the part of interest of the source schema
3. The target schema misses some constructs to render some meaning of interest in the source schema
4. The target schema contains constructs that overlap with but do not cover a certain source construct.

Case (a) is trivially met by considering only the part of interest of the source schema respectively. This is characteristic for data provision, because most local systems contain some data not of interest for the target aggregator.

Case (b) can be described in terms of a “query containment condition”: When data are transformed into the target schema B, for any answer to a query possible under schema A, there should exist a query possible under schema B that returns an answer set that semantically comprises the respective query answer under A.

As illustration, let us consider:

Schema A: Person – attended -> Course, Person – taught -> Course.

Schema B: Actor – participated in -> Meeting.

Mapping: Person -> Actor

Course -> Meeting

attended -> participated in

taught -> participated in.

An instance in A: “George(Person) taught CRM Course(Course)”

transforms into B: “George(Actor) participated in CRM Course(Meeting)”

An instance in A: “Gerald(Person) attended CRM Course(Course)”

transforms into B: “Gerald(Actor) participated in CRM Course(Meeting)”

Querying “participated in” in schema B returns:

“George(Actor) participated in CRM Course(Meeting)”

“Gerald(Actor) participated in CRM Course(Meeting)”,

without distinguishing “taught” from “attended” as done in schema A.

The effect of such a mapping is that all facts described by schema A can be found by querying the mapped data under schema B, but the facts may be described less precisely. Not all facts that can be distinguished by querying schema A may be distinguished also by querying schema B. This situation is the ideal case for information integration under a global schema, which cannot be expected to foresee all details of all sources to come. It can be described as a preference of “recall over precision” (see chapters below).

Recognizing such a mapping condition is intellectually the same kind of insight as that in steps 6 and 7 in Phase B, only with the difference, that the target classes and properties are already given and not invented or detected in the process. The user must decide, if the given classes and properties of the target contain adequate equivalents or generalizations, or if some additional classes or properties would be needed to be invented or can be taken from other models, not excluding the source itself, in order to cover the ability to map the source.

Nevertheless, concepts from schema A are missing in schema B. A “trick” helps to overcome this problem, even without requiring an extension to schema B: add to classes a “type” and to properties a “role”.

Then,

an instance in A: “George(Person) taught CRM Course(Course)”

transforms into B: “George(Actor, type: Person) participated in(role: taught)

CRM Course(Meeting, type: Course)”

An instance in A: “Gerald(Person) attended CRM Course(Course)”

transforms into B: “Gerald(Actor, type: Person) participated in(role: attended)

CRM Course(Meeting, type: Course)”

The two generic properties “has type” and “in role” allow for turning source classes “into data”. Then, a loss-free mapping is accomplished. This works long as roles and types are not related. In schema A2: Student – attended -> Course, Professor – taught -> Course, the mapping to B would still work, but the “binding” of “taught” to “Professor” cannot be rendered by adding types and roles as data. If this is needed on the target side, it would require an extension.

In the following, we distinguish two fundamental use cases: the maintenance of the target schema foresees user-driven extensions, or not. In the latter case, nothing can be done about a true mismatch (case (c) above). In case (d), overlapping constructs, the best overlapping may be chosen as a surrogate for a true generalization. The mapping job is finished with steps 6 and 7 of the development process above.

If, on the other side, extensions are foreseen—this is the default for using the CRM methodology—the source schema and data encoded under it are seen as empirical source material and we go back to step 1 of the development process if the source schema has not been sufficiently mapped for the purpose. For any specific construct to be added, and for overlapping concepts (case (d) above), suitable generalizations should be considered. In the CRM Methodology we foresee in particular that also existing concepts in the target should be considered to be widened in scope when a particular mapping provides evidence that this is useful for the purpose of the target.

As long as the target schema provides suitable generalizations, the extension is regarded to be “compatible” and can stay in the use of a local community, regardless of whether or not it is proposed as a recommendation for a wider user community. If, however, it requires generalizations not covered by the target, it is preferable to modify the target itself. As seen through this lens, mapping is a special case of an ontology development process, especially, if the target foresees extension and is open to ongoing modification.

# Principles Introduction

This part of the guide provides a set of individual principles as a reference to be taken into account alongside the Process Model for building and/or extending [ontologies](#_vj2xbzdro28i) for integrating data in [empirical domains](#_jwb1cfff8b8). The principles are organized according to eight thematic areas which each group together a series of related methodological principles and recommendations.

These thematic groups are:

1. Engineering from an empirical base
2. Knowledge structure basics
3. Concept relevance
4. Open world
5. Open world and knowledge progress
6. Open world and knowledge bases
7. Objectivity
8. Language & Concepts

For each of the overarching themes there is an introduction followed by a detailed description of the principles for that thematic group.

At the end of the guide, the principles are reordered in a processual checklist which is intended to be used as a reminder and verification tool during the setup, execution and evaluation of a modelling activity.

That being said, the principles introduced here can also be taken up and applied under a number of different considerations and use scenarios beyond a direct modelling exercise. Particularly they bear a relation to questions of:

**General Ontological Methods (OM)**

The principles can be applied to the general question of how to understand and then represent reality in formal ontological terms by means of the identification and declaration of formal relations between identifiable classes. The question approached here is: what are the correct means and limits of the representation of reality by a formal ontological construction?

**Ontology Use (OU)**

In implementing a formal ontology in an actual [information system](#_rmj4f9tw5j0v), care must be taken in the translation of the ontology within the technical environment. The principles provide guidance how to make this translation successfully.

**Knowledge Base/Representation Principles (KB)**

The principles intend to inform on the best technical means of representing knowledge and the ability and likelihood to know what are or were stated/known in terms of formal propositions at some time about the world.

**CIDOC CRM Method (CM)**

The principles provide a guideline to building a formal ontology that can act as a standard. In this regard, the CIDOC CRM in particular is considered and is referenced as an ISO standard which models cultural-historical discourse, including all scientific activities, focusing on a material level of description.

**Principles for Building a Standard (PS)**

In general, the principles can inform decisions on establishing manageable [units of documentation](#_y6ql6eualny6). These enable information integration for some domain but also ensure the long-term robustness of the declared classes and relations to support [monotonic](#_r0bgjwaqhz6k) (backward-compatible) revisions of the ontology.

Therefore, each principle given in its main form as a solution can potentially be further elaborated with regards to these various aspects (not carried out in this document).

# Glossary

**binary relationship**

A relationship between exactly two individual entities, such as one person “is parent of” another person. Some knowledge representation languages, such as RDF, support only binary relationships, known as “properties”. Any more complex relationship is then described as a class connecting three or more properties, e.g., to describe a temporary membership in a group. Other languages explicitly support relationships between multiple individuals, such as the Entity-Relationship Model.

**class**

A category of items that share one or more common traitsserving as criteria to identify the items belonging to the class. These **properties** need not be explicitly formulated in logical terms, but may be described in a text (here called a **scope note**) that refers to a common conceptualisation of domain experts. The sum of these traits is called the **intension** of the class. A class may be the **domain** or **range** of none, one or more properties formally defined in a model. The formally defined properties need not be part of the intension of their domains or ranges: such properties are optional. An item that belongs to a class is called an **instance** of this class. A class is associated with an open set of real-life instances, known as the **extension** of the class. Here “open” is used in the sense that it is generally beyond our capabilities to know all instances of a class in the world and indeed that the future may bring new instances about at any time (**Open World**). Therefore, a class cannot be defined by enumerating its instances. A class plays a role analogous to a grammatical noun and can be completely defined without reference to any other construct (unlike properties**,** which must have an unambiguously defined domain and range). In some contexts, the terms individual class, entity or node are used synonymously with class. A class is a universal.

**closed world**

Describes information systems which assume that the information stored in them is complete relative to the universe of discourse they intend to describe. In particular, statements that cannot be shown to be true are regarded as false. Quod non est in actis non est in mundo.

**complement**

A complement of a class A with respect to one of its superclasses B consists of all instances of its superclass B which are not instances of the class A.

**compression**

A term first used by Fauconnier and Turner (2002)[[4]](#footnote-4) to describe a function of our conscious thinking that reduces complex relationships to seemingly simple ones for particular contexts without losing the ability to recover the full meaning from the subconscious.

**data**

Encoded information filled into the fields of a data structure for use in a formal way by algorithms and machines, typically managed in an information system in digital format.

**data structure**

Encoded predefined set of typed data fields and rules for their allowed arrangement, which is used as classification and expression of relationships of the data entered into the fields. It is also intended to instruct an information system on how to manage data, typically associated with the fields in a database entry form (the schema of the database).

**concept, primitive**

A class or relation which is not exhaustively defined in terms of logical expressions combining other concepts. For instance: “E21 Person”.

**concept, derived**

A class or relation which is exhaustively defined in terms of logical expressions combining other concepts. For instance: “mother = female AND Person AND has child”.

**empirical domain**

Domain of scholarship which relies on observations to produce new knowledge. We take this here in a very wide sense, which includes historical documents as observable items and the observations reported in documents, albeit critical about them.

**information**

In the narrower sense meant here: a structured set of symbols compiled by some actor who knows to resolve the meaning of the symbols and intended as a message to other actors supposed to be able to resolve the meaning of the symbols in the same way. It could be as general as texts, but also graphics and in particular data. By resolving the symbols, an actor may turn information into knowledge according to their trust in it.

**information structure**

Any form of grammar needed to interpret the meaning of the relative position of a symbol (or word) in an information unit.

**information source**

Any document or system containing units of information.

**information system**

A computer-based system that allows users to access and communicate information provided by multiple users, without the need of human mediators knowing content. Moreover, an information system allows to reasonably relate information by means of logical operations in a way useful for their business and in ways not previously obvious to the individual providers.

**information value**

An element or encoded fact in a particular information unit, in contrast to a (reusable) information structure.

**inheritance**

Describes the fact that the properties of a superclass are also properties of its

subclasses.

**instance, class**

An item that can be characterized by a particular class.

**instance, relation**

A factual relationship of a type given by a specific property, between the instance of the domain class and the instance of the range class of this property.

**intension**

The intension of a **class** or **property** is its intended meaning. It consists of one or more common traitsshared by all **instances** of the class or property. These traits need not be explicitly formulated in logical terms, but may just be described in a text (here called a **scope note**) that refers to a conceptualisation common to domain experts. In particular the so-called **primitive** concepts, which make up most of the CRM, cannot be further reduced to other concepts by logical terms.

**knowledge base**

A database that can manage data formatted by a so-called knowledge representation language, such as KL-ONE, TELOS, KIF, DAML, OIL, RDF(S), OWL etc.

**knowledge representation**

Structuring of information by a so-called knowledge representation language, which is supposed to be more similar to the way humans think and the way in which reality is perceived as containing discrete thing than other data encoding paradigms (e.g., E-R, XML).

**monotonicity**

Describes the case in which the relations between classes in an ontology remain valid even if (suitable) new classes and relations are added.

**ontology or ontological model**

Formal naming and definition of the classes and relations of the entities (the stuff) that exist for a domain and can unambiguously be shared in a community of users.

**open world**

Characterizes information systems which manage data consistently with the assumption that the information stored in them is incomplete relative to the universe of discourse they intend to describe. In particular, missing information in the system is not interpreted as non-existence of the respective properties in the universe of discourse.

**particular**

An item that does not have instances. Often an instance of a class.

**perdurant**

Items depending on time. They can only be captured adequately in relation to passing time (e.g. in a video).

**polysemy**

Describes the capacity of a word to have many meanings.

**reality**

We regard reality as that which is unique in space and time and makes independent observations potentially comparable about their reference, including mental states of humans.

**relation or property**

A relation defines a link of a specific kind between two classes. The origin class is the domain and the destination class is the range of the relation. A relation is a universal.

**state of affairs**

A representation of an aspect of reality by a set of relationships holding for some time span.

**subclass**

A class that is a specialization of one or more classes (its **superclasses**), i.e. all instances of the subclass are also instances of the superclasses.

**superclass**

A class that is a generalization of one or more classes (its **subclasses**), i.e. all instances of all its subclasses are also instances of the superclass.

**tautology**

A proposition that is true in every possible interpretation and therefore cannot lead to new knowledge.

**universal**

An entity that has instances. Classes and properties are universals.

# Engineering from an Empirical Base

The creation of functional integrative ontologies depends on a 'bottom up' strategy of working from real empirical information - data and corresponding [data structures](#_25e6qi7eafg) - in order to abstract relevant [relations](#_xlqxrujni66b) and [classes](#_s5n1e7queg2k) that will adequately cover the modelled domain. By adhering to an evidence-based approach, the conceptual modeller is able to build a model that is capable of providing an explanation/translation of relevant information from target data structures in the domain into a common model. The ultimate criterion for adjudicating such adequacy is the ability of the resultant model to enable scholars/scientists to pose and find answers to their research questions via data described in terms of the proposed model. Therefore, an essential part of the empirical evidence to be gathered includes the high-level research questions that scholars/scientists aim to answer via their data collection. These questions form the necessary contextual basis for understanding and modelling data and conceptualizations.

Under this topic, we identify three principles:

* 1. [Model from existing actually used structured information sources (whenever available)](#_2et92p0)

[1.2 Model according to the research questions justifying the structured information](#_tyjcwt)

[1.3 Model from actual information values](#_3dy6vkm)

## 1.1 Model from existing / actually used structured information sources (whenever available)

|  |  |  |
| --- | --- | --- |
| ID | Principle | Slogan |
| 1.1 | Model from existing / actually used structured information sources (whenever available) | Models should be useful |
| Problem Description | | |
| What is proper source material for my ontology? | | |
| Argument / Solution | | |
| Modelling from existing, actually used structured information (e.g.: databases, spreadsheets, RDF documents, XML documents, structured analogue documents, etc.) ensures that the underlying concepts revealed by the process are useful. It establishes that the model models information that people actually devote resources to encode. Where no existing structured [information sources](#_masqkz62ov4w) exist to begin modelling from, intuitive sketches of potential structured information sources from the target community can be used as an empirical information source. | | |
| ☺ Eg. | Modelling CRMarcheo from national excavation recording forms | |
| ☹ Eg. | Modelling FRBRoo from FRBR, which introduced an intended practice intertwined with existing documentation practice | |

|  |
| --- |
| Applicability |
| OM/CM/PS |

## 1.2 Model according to the research questions justifying the structured information

|  |  |  |
| --- | --- | --- |
| ID | Principle | Slogan |
| 1.2 | Model according to the research questions justifying the structured information | Why do you need this field? |
| Problem Description | | |
| How can we determine accurately the semantic interpretation needed from potentially ambiguous or overdetermined information sources? | | |
| Argument / Solution | | |
| Structured information can have many senses. The relation to the actual use of the information can be quite intuitive. The ontological interpretation must follow the real research questions for which the information is used or can be used. This requires the elicitation of (sometimes implicit) research questions from the domain users/community by interview. | | |
| ☺ Eg. | A field "age" in the CIDOC Relational Model (precursor to CIDOC CRM). This field was fundamentally ambiguous and its use/content could not be understood without reference to interviews with researchers who indicated that it was meant to describe the life phase in which a biological specimen was killed. This interpretation guided the modelling to an unambiguous semantic expression: "Life stage type". Same field age was used by art historians to describe role of artistic artefact in the process of a creating a work.  Also: common meaning behind archaeological "find", biological "occurrence", or archaeological "prototype", biological "holotype" | |
| ☹ Eg. |  | |

|  |
| --- |
| Applicability |
|  |

## 1.3 Model from actual information values

|  |  |  |
| --- | --- | --- |
| ID | Principle | Slogan |
| 1.3 | Model from actual [information values](#_vdys77uawz7p) | Model only real cases, Model what is meant not what is thought to be meant |
| Problem Description | | |
| Can we model from the bare [information structure](#_1uyw0o2kz95x) (or do we need to have instantiated information values)? | | |
| Argument / Solution | | |
| When modelling the information of interest to represent for researchers it is important to go beyond the intention of the bare information structures (such as data forms in an information system) to see how they are actually put to use. Information structures such as data forms represent intuitions, local information goals, and practical constraints. They are an intention limited by circumstances. Information actually gathered and entered into structures provides evidence beyond intention, revealing actual practice. Actual practice connects to the real world, reveals exceptions, ambiguities in information structure definitions; it shows not “how it is supposed to be done” but “how it is done”. If in the modelling project in question no information values are yet available, solicit information value samples from researchers. | | |
| ☺ Eg. | From the actual data one could observe thet the field “Age” in the CIDOC Relational Model was used to describe the products of phases of an artistic process: i.e., sketch, underdrawing, etc.  Use of a Field Sex: Used both for ‘M/F’ and ‘Yes/No’ values | |
| ☹ Eg. | Take "object name" in a collection management system for a proper name | |

|  |
| --- |
| Applicability |
| OM/ CM/PS |

1. Universal Modelling Language, used for object oriented computer science modelling. See http://www.uml.org (checked 2018-05-11) [↑](#footnote-ref-1)
2. For instance, weight of a living being is in constant flux. Length of wood changes with humidity. A coast line is both a fractal penetration of water and land and dependent on waves and sea level. A car crash begins between a driver’s mistake and touching metal. [↑](#footnote-ref-2)
3. Aldo Gangemi 2006 [↑](#footnote-ref-3)
4. FAUCONNIER, G., AND TURNER, M., 2002. *The Way we Think : Conceptual Blending and the Mind’s Complexities*, Basic Books, New York. [↑](#footnote-ref-4)