**Ontology (information science)**

From Wikipedia, the free encyclopedia

Jump to: [navigation](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#mw-head), [search](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#p-search)

In [computer science](http://en.wikipedia.org/wiki/Computer_science) and [information science](http://en.wikipedia.org/wiki/Information_science), an **ontology** formally represents knowledge as a set of concepts within a [domain](http://en.wikipedia.org/wiki/Domain_of_discourse), and the relationships between those concepts. It can be used to [reason](http://en.wikipedia.org/wiki/Reasoning) about the entities within that domain and may be used to describe the domain.

In theory, an ontology is a "formal, explicit specification of a shared conceptualisation".[[1]](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#cite_note-TRG93-0) An ontology renders shared vocabulary and taxonomy which models a domain with the definition of objects and/or concepts and their properties and relations.[[2]](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#cite_note-1)

Ontologies are the structural frameworks for organizing information and are used in [artificial intelligence](http://en.wikipedia.org/wiki/Artificial_intelligence), the [Semantic Web](http://en.wikipedia.org/wiki/Semantic_Web), [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering), [software engineering](http://en.wikipedia.org/wiki/Software_engineering), [biomedical informatics](http://en.wikipedia.org/wiki/Biomedical_informatics), [library science](http://en.wikipedia.org/wiki/Library_science), [enterprise bookmarking](http://en.wikipedia.org/wiki/Enterprise_bookmarking), and [information architecture](http://en.wikipedia.org/wiki/Information_architecture) as a form of [knowledge representation](http://en.wikipedia.org/wiki/Knowledge_representation) about the world or some part of it. The creation of domain ontologies is also fundamental to the definition and use of an [enterprise architecture framework](http://en.wikipedia.org/wiki/Enterprise_architecture_framework).

|  |
| --- |
| **Contents*** [1 Overview](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#Overview)
* [2 History](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#History)
* [3 Ontology components](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#Ontology_components)
* [4 Domain ontologies and upper ontologies](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#Domain_ontologies_and_upper_ontologies)
* [5 Ontology engineering](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#Ontology_engineering)
* [6 Ontology languages](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#Ontology_languages)
* [7 Examples of published ontologies](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#Examples_of_published_ontologies)
* [8 Ontology libraries](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#Ontology_libraries)
* [9 Examples of applications using ontology engines](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#Examples_of_applications_using_ontology_engines)
* [10 See also](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#See_also)
* [11 References](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#References)
* [12 Further reading](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#Further_reading)
* [13 External links](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#External_links)
 |

**Overview**

The term [*ontology*](http://en.wikipedia.org/wiki/Ontology) has its origin in [philosophy](http://en.wikipedia.org/wiki/Philosophy) and has been applied in many different ways. The word element [*onto-*](http://en.wiktionary.org/wiki/onto-) comes from the [Greek](http://en.wikipedia.org/wiki/Greek_language) [*ὤν*](http://en.wiktionary.org/wiki/%E1%BD%A4%CE%BD)*, ὄντος* « being; that which is », present participle of the verb [*εἰμί*](http://en.wiktionary.org/wiki/%CE%B5%E1%BC%B0%CE%BC%CE%AF) « be ». The core meaning within [computer science](http://en.wikipedia.org/wiki/Computer_science) is a model for describing the world that consists of a set of types, properties, and relationship types. Exactly what is provided around these varies, but they are the essentials of an ontology. There is also generally an expectation that there be a close resemblance between the real world and the features of the model in an ontology.[[3]](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#cite_note-2)

What many ontologies have in common in both computer science and in philosophy is the representation of entities, ideas, and events, along with their properties and relations, according to a system of categories. In both fields, one finds considerable work on problems of ontological relativity (e.g., [Quine](http://en.wikipedia.org/wiki/Willard_Van_Orman_Quine) and [Kripke](http://en.wikipedia.org/wiki/Saul_Kripke) in philosophy, [Sowa](http://en.wikipedia.org/wiki/John_F._Sowa) and [Guarino](http://en.wikipedia.org/wiki/Nicola_Guarino) in computer science),[[4]](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#cite_note-3), and debates concerning whether a [normative](http://en.wikipedia.org/wiki/Normative) ontology is viable (e.g., debates over [foundationalism](http://en.wikipedia.org/wiki/Foundationalism) in philosophy, debates over the [Cyc](http://en.wikipedia.org/wiki/Cyc) project in AI). Differences between the two are largely matters of focus. Philosophers are less concerned with establishing fixed, controlled vocabularies than are researchers in computer science, while computer scientists are less involved in discussions of first principles, such as debating whether there are such things as [fixed essences](http://en.wikipedia.org/w/index.php?title=Fixed_essences&action=edit&redlink=1) or whether entities must be ontologically more primary than processes.

Other fields make ontological assumptions that are sometimes explicitly elaborated and explored. For instance, the [definition and ontology of economics](http://en.wikipedia.org/wiki/Philosophy_and_economics#definition_and_ontology_of_economics) (also sometimes called the [political economy](http://en.wikipedia.org/wiki/Political_economy)) is hotly debated especially in [Marxist economics](http://en.wikipedia.org/wiki/Marxist_economics)[[1]](http://cje.oxfordjournals.org/content/31/4/539.short) where it is a primary concern, but also in other subfields [[2]](http://ideas.repec.org/p/pra/mprapa/5566.html). Such concerns intersect with those of information science when the intent of a simulation or model is to enable decisions in the economic realm, for instance, to determine what [capital assets](http://en.wikipedia.org/wiki/Capital_asset) are at risk or how much (see [risk management](http://en.wikipedia.org/wiki/Risk_management)). All social sciences have explicit ontology issues because they do not have hard [falsification](http://en.wikipedia.org/wiki/Falsification) criteria like most models in physical sciences - indeed the lack of such widely accepted hard falsification criteria is what defines a social or soft science.

**History**

Historically, ontologies arise out of the branch of [philosophy](http://en.wikipedia.org/wiki/Philosophy) known as [metaphysics](http://en.wikipedia.org/wiki/Metaphysics), which deals with the nature of reality – of what exists. This fundamental branch is concerned with analyzing various types or modes of existence, often with special attention to the relations between [particulars](http://en.wikipedia.org/wiki/Particular) and [universals](http://en.wikipedia.org/wiki/Universal_%28metaphysics%29), between [intrinsic and extrinsic properties](http://en.wikipedia.org/wiki/Intrinsic_and_extrinsic_properties_%28philosophy%29), and between [essence](http://en.wikipedia.org/wiki/Essence) and [existence](http://en.wikipedia.org/wiki/Existence). The traditional goal of ontological inquiry in particular is to divide the world "at its joints" to discover those fundamental categories or kinds into which the world’s objects naturally fall.[[5]](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#cite_note-PCB94-4)

During the second half of the 20th century, philosophers extensively debated the possible methods or approaches to building ontologies without actually *building* any very elaborate ontologies themselves. By contrast, computer scientists were building some large and robust ontologies, such as [WordNet](http://en.wikipedia.org/wiki/WordNet) and [Cyc](http://en.wikipedia.org/wiki/Cyc), with comparatively little debate over *how* they were built.

Since the mid-1970s, researchers in the field of [artificial intelligence](http://en.wikipedia.org/wiki/Artificial_intelligence) (AI) have recognized that capturing knowledge is the key to building large and powerful AI systems. AI researchers argued that they could create new ontologies as [computational models](http://en.wikipedia.org/wiki/Computational_model) that enable certain kinds of [automated reasoning](http://en.wikipedia.org/wiki/Automated_reasoning). In the 1980s, the AI community began to use the term *ontology* to refer to both a theory of a modeled world and a component of knowledge systems. Some researchers, drawing inspiration from philosophical ontologies, viewed computational ontology as a kind of applied philosophy.[[6]](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#cite_note-TG08-5)

In the early 1990s, the widely cited Web page and paper "Toward Principles for the Design of Ontologies Used for Knowledge Sharing" by [Tom Gruber](http://en.wikipedia.org/wiki/Tom_Gruber)[[7]](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#cite_note-TRG95-6) is credited with a deliberate definition of *ontology* as a technical term in [computer science](http://en.wikipedia.org/wiki/Computer_science). Gruber introduced the term to mean a specification of a conceptualization. That is, "An ontology is a description (like a formal specification of a program) of the concepts and relationships that can formally exist for an agent or a community of agents. This definition is consistent with the usage of ontology as set of concept definitions, but more general. And it is a different sense of the word than its use in philosophy".[[8]](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#cite_note-TRG01-7)

According to Gruber (1993), "Ontologies are often equated with taxonomic hierarchies of classes, class definitions, and the subsumption relation, but ontologies need not be limited to these forms. Ontologies are also not limited to [conservative definitions](http://en.wikipedia.org/wiki/Conservative_extension) — that is, definitions in the traditional logic sense that only introduce terminology and do not add any knowledge about the world.[[9]](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#cite_note-8) To specify a conceptualization, one needs to state axioms that do constrain the possible interpretations for the defined terms."[[1]](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#cite_note-TRG93-0)

**Ontology components**

*Main article:* [*Ontology components*](http://en.wikipedia.org/wiki/Ontology_components)

Contemporary ontologies share many structural similarities, regardless of the language in which they are expressed. As mentioned above, most ontologies describe individuals (instances), classes (concepts), attributes, and relations. In this section each of these components is discussed in turn.

Common components of ontologies include:

* Individuals: instances or objects (the basic or "ground level" objects)
* [Classes](http://en.wikipedia.org/wiki/Class_%28set_theory%29): [sets](http://en.wikipedia.org/wiki/Class_%28set_theory%29), collections, concepts, [classes in programming](http://en.wikipedia.org/wiki/Class_%28computer_science%29), [types of objects](http://en.wikipedia.org/wiki/Class_%28philosophy%29), or kinds of things
* [Attributes](http://en.wikipedia.org/wiki/Attribute_%28computing%29): aspects, properties, features, characteristics, or parameters that objects (and classes) can have
* [Relations](http://en.wikipedia.org/wiki/Relation_%28mathematics%29): ways in which classes and individuals can be related to one another
* Function terms: complex structures formed from certain relations that can be used in place of an individual term in a statement
* Restrictions: formally stated descriptions of what must be true in order for some assertion to be accepted as input
* Rules: statements in the form of an if-then (antecedent-consequent) sentence that describe the logical inferences that can be drawn from an assertion in a particular form
* Axioms: assertions (including rules) in a [logical form](http://en.wikipedia.org/wiki/Logical_form) that together comprise the overall theory that the ontology describes in its domain of application. This definition differs from that of "axioms" in [generative grammar](http://en.wikipedia.org/wiki/Generative_grammar) and [formal logic](http://en.wikipedia.org/wiki/Formal_logic). In those disciplines, axioms include only statements asserted as *a priori* knowledge. As used here, "axioms" also include the theory derived from axiomatic statements
* [Events](http://en.wikipedia.org/wiki/Event_%28philosophy%29): the changing of attributes or relations

Ontologies are commonly encoded using [ontology languages](http://en.wikipedia.org/wiki/Ontology_language).

**Domain ontologies and upper ontologies**

A domain ontology (or domain-specific ontology) models a specific domain, which represents part of the world. Particular meanings of terms applied to that domain are provided by domain ontology. For example the word [*card*](http://en.wiktionary.org/wiki/card) has many different meanings. An ontology about the domain of [poker](http://en.wikipedia.org/wiki/Poker) would model the "[playing card](http://en.wikipedia.org/wiki/Playing_card)" meaning of the word, while an ontology about the domain of [computer hardware](http://en.wikipedia.org/wiki/Computer_hardware) would model the "[punched card](http://en.wikipedia.org/wiki/Punched_card)" and "[video card](http://en.wikipedia.org/wiki/Video_card)" meanings.

An [upper ontology](http://en.wikipedia.org/wiki/Upper_ontology_%28computer_science%29) (or foundation ontology) is a model of the common objects that are generally applicable across a wide range of domain ontologies. It employs a [core glossary](http://en.wikipedia.org/wiki/Core_glossary) that contains the terms and associated object descriptions as they are used in various relevant domain sets. There are several standardized upper ontologies available for use, including [Dublin Core](http://en.wikipedia.org/wiki/Dublin_Core), [GFO](http://en.wikipedia.org/wiki/General_Formal_Ontology), [OpenCyc](http://en.wikipedia.org/wiki/Cyc#OpenCyc)/[ResearchCyc](http://en.wikipedia.org/wiki/Cyc%22%20%5Cl%20%22ResearchCyc%22%20%5Co%20%22Cyc), [SUMO](http://en.wikipedia.org/wiki/Suggested_Upper_Merged_Ontology), and [DOLCE](http://en.wikipedia.org/w/index.php?title=Descriptive_Ontology_for_Linguistic_and_Cognitive_Engineering&action=edit&redlink=1).[[10]](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#cite_note-DOLCE-9) [WordNet](http://en.wikipedia.org/wiki/WordNet), while considered an upper ontology by some, is not strictly an ontology. However, it has been employed as a linguistic tool for learning domain ontologies.[[11]](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#cite_note-10)

The [Gellish](http://en.wikipedia.org/wiki/Gellish) ontology is an example of a combination of an upper and a domain ontology.

Since domain ontologies represent concepts in very specific and often eclectic ways, they are often incompatible. As systems that rely on domain ontologies expand, they often need to merge domain ontologies into a more general representation. This presents a challenge to the ontology designer. Different ontologies in the same domain can also arise due to different perceptions of the domain based on cultural background, education, ideology, or because a different representation language was chosen.

At present, merging ontologies that are not developed from a common foundation ontology is a largely manual process and therefore time-consuming and expensive. Domain ontologies that use the same foundation ontology to provide a set of basic elements with which to specify the meanings of the domain ontology elements can be merged automatically. There are studies on generalized techniques for merging ontologies[[*citation needed*](http://en.wikipedia.org/wiki/Wikipedia%3ACitation_needed)], but this area of research is still largely theoretical.

**Ontology engineering**

*Main article:* [*Ontology engineering*](http://en.wikipedia.org/wiki/Ontology_engineering)

[Ontology engineering](http://en.wikipedia.org/wiki/Ontology_engineering) (or ontology building) is a subfield of [knowledge engineering](http://en.wikipedia.org/wiki/Knowledge_engineering) that studies the methods and methodologies for building ontologies. It studies the ontology development process, the ontology life cycle, the methods and methodologies for building ontologies, and the tool suites and languages that support them.[[12]](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#cite_note-PFC04-11)[[13]](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#cite_note-DMN-12)

Ontology engineering aims to make explicit the knowledge contained within software applications, and within enterprises and business procedures for a particular domain. Ontology engineering offers a direction towards solving the interoperability problems brought about by semantic obstacles, such as the obstacles related to the definitions of business terms and software classes. Ontology engineering is a set of tasks related to the development of ontologies for a particular domain.[[14]](http://en.wikipedia.org/wiki/Ontology_%28information_science%29#cite_note-PIS00-13)

Ontology components

From Wikipedia, the free encyclopedia

Jump to: [navigation](http://en.wikipedia.org/wiki/Ontology_components%22%20%5Cl%20%22mw-head), [search](http://en.wikipedia.org/wiki/Ontology_components%22%20%5Cl%20%22p-search)

|  |  |
| --- | --- |
| http://upload.wikimedia.org/wikipedia/en/thumb/9/99/Question_book-new.svg/50px-Question_book-new.svg.png | This article **needs additional** [**citations**](http://en.wikipedia.org/wiki/Wikipedia%3ACiting_sources#Inline_citations) **for** [**verification**](http://en.wikipedia.org/wiki/Wikipedia%3AVerifiability). Please help [improve this article](http://en.wikipedia.org/w/index.php?title=Ontology_components&action=edit) by adding citations to [reliable sources](http://en.wikipedia.org/wiki/Wikipedia%3AIdentifying_reliable_sources). Unsourced material may be [challenged](http://en.wikipedia.org/wiki/Template%3ACitation_needed) and [removed](http://en.wikipedia.org/wiki/Wikipedia%3AVerifiability#Burden_of_evidence). *(June 2009)* |

Contemporary [ontologies](http://en.wikipedia.org/wiki/Ontology_%28information_science%29) share many structural similarities, regardless of the language in which they are expressed. Most ontologies describe individuals (instances), classes (concepts), attributes, and relations.

|  |
| --- |
| **Contents**[[hide](http://en.wikipedia.org/wiki/Ontology_components)] * [1 Overview](http://en.wikipedia.org/wiki/Ontology_components#Overview)
* [2 Individuals](http://en.wikipedia.org/wiki/Ontology_components#Individuals)
* [3 Classes](http://en.wikipedia.org/wiki/Ontology_components#Classes)
* [4 Attributes](http://en.wikipedia.org/wiki/Ontology_components#Attributes)
* [5 Relationships](http://en.wikipedia.org/wiki/Ontology_components#Relationships)
* [6 References](http://en.wikipedia.org/wiki/Ontology_components#References)
 |

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Ontology_components&action=edit&section=1)**] Overview**

Common components of ontologies include:

* Individuals: instances or objects (the basic or "ground level" objects)
* [Classes](http://en.wikipedia.org/wiki/Class_%28set_theory%29): [sets](http://en.wikipedia.org/wiki/Set_%28computer_science%29), collections, concepts, types of objects, or kinds of things.[[1]](http://en.wikipedia.org/wiki/Ontology_components#cite_note-0)
* [Attributes](http://en.wikipedia.org/wiki/Attribute_%28computing%29): aspects, properties, features, characteristics, or parameters that objects (and classes) can have
* [Relations](http://en.wikipedia.org/wiki/Relation_%28mathematics%29): ways in which classes and individuals can be related to one another
* Function terms: complex structures formed from certain relations that can be used in place of an individual term in a statement
* Restrictions: formally stated descriptions of what must be true in order for some assertion to be accepted as input
* Rules: statements in the form of an if-then (antecedent-consequent) sentence that describe the logical inferences that can be drawn from an assertion in a particular form
* Axioms: assertions (including rules) in a [logical form](http://en.wikipedia.org/wiki/Logical_form) that together comprise the overall theory that the ontology describes in its domain of application. This definition differs from that of "axioms" in generative grammar and formal logic. In these disciplines, axioms include only statements asserted as *a priori* knowledge. As used here, "axioms" also include the theory derived from axiomatic statements.
* [Events](http://en.wikipedia.org/wiki/Event_%28philosophy%29): the changing of attributes or relations

Ontologies are commonly encoded using [ontology languages](http://en.wikipedia.org/wiki/Ontology_language).

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Ontology_components&action=edit&section=2)**] Individuals**

Individuals (instances) are the basic, "ground level" components of an ontology. The individuals in an ontology may include concrete objects such as people, animals, tables, automobiles, molecules, and planets, as well as abstract individuals such as numbers and words (although there are differences of opinion as to whether numbers and words are classes or individuals). Strictly speaking, an ontology need not include any individuals, but one of the general purposes of an ontology is to provide a means of classifying individuals, even if those individuals are not explicitly part of the ontology.

In formal extensional ontologies, only the utterances of words and numbers are considered individuals – the numbers and names themselves are classes. In a 4D ontology, an individual is identified by its spatio-temporal extent. Examples of formal extensional ontologies are [ISO 15926](http://en.wikipedia.org/wiki/ISO_15926) and the model in development by the [IDEAS Group](http://en.wikipedia.org/wiki/IDEAS_Group).

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Ontology_components&action=edit&section=3)**] Classes**

Classes – concepts that are also called *type*, *sort*, *category*, and *kind* – can be defined as an extension or an intension. According to an extensional definition, they are abstract groups, sets, or collections of objects. According to an intensional definition, they are abstract objects that are defined by values of aspects that are constraints for being member of the class. The first definition of class results in ontologies in which a class is a subclass of collection. The second definition of class results in ontologies in which collections and classes are more fundamentally different. Classes may classify individuals, other classes, or a combination of both. Some examples of classes:[[2]](http://en.wikipedia.org/wiki/Ontology_components#cite_note-1)

* *Person*, the class of all people, or the abstract object that can be described by the criteria for being a person.
* *Vehicle*, the class of all vehicles, or the abstract object that can be described by the criteria for being a vehicle.
* *Car*, the class of all cars, or the abstract object that can be described by the criteria for being a car.
* *Class*, representing the class of all classes, or the abstract object that can be described by the criteria for being a class.
* *Thing*, representing the class of all things, or the abstract object that can be described by the criteria for being a thing (and not nothing).

Ontologies vary on whether classes can contain other classes, whether a class can belong to itself, whether there is a universal class (that is, a class containing everything), etc. Sometimes restrictions along these lines are made in order to avoid certain well-known [paradoxes](http://en.wikipedia.org/wiki/Paradox).

The classes of an ontology may be [extensional](http://en.wikipedia.org/wiki/Extensional) or [intensional](http://en.wikipedia.org/wiki/Intensional) in nature. A class is extensional if and only if it is characterized solely by its membership. More precisely, a class C is extensional if and only if for any class C', if C' has exactly the same members as C, then C and C' are identical. If a class does not satisfy this condition, then it is intensional. While extensional classes are more well-behaved and well-understood mathematically, as well as less problematic philosophically, they do not permit the fine grained distinctions that ontologies often need to make. For example, an ontology may want to distinguish between the class of all creatures with a kidney and the class of all creatures with a heart, even if these classes happen to have exactly the same members. In most upper ontologies, the classes are defined intensionally. Intensionally defined classes usually have necessary conditions associated with membership in each class. Some classes may also have sufficient conditions, and in those cases the combination of necessary and sufficient conditions make that class a fully *defined* class.

Importantly, a class can subsume or be subsumed by other classes; a class subsumed by another is called a *subclass* (or *subtype*) of the subsuming class (or *supertype*). For example, *Vehicle* subsumes *Car*, since (necessarily) anything that is a member of the latter class is a member of the former. The subsumption relation is used to create a hierarchy of classes, typically with a maximally general class like *Anything* at the top, and very specific classes like *2002 Ford Explorer* at the bottom. The critically important consequence of the subsumption relation is the inheritance of properties from the parent (subsuming) class to the child (subsumed) class. Thus, anything that is necessarily true of a parent class is also necessarily true of all of its subsumed child classes. In some ontologies, a class is only allowed to have one parent (*single inheritance*), but in most ontologies, classes are allowed to have any number of parents (*multiple inheritance*), and in the latter case all necessary properties of each parent are inherited by the subsumed child class. Thus a particular class of animal (*HouseCat*) may be a child of the class *Cat* and also a child of the class *Pet*.



A partition is a set of related classes and associated rules that allow objects to be classified by the appropriate subclass. The rules correspond with the aspect values that distinguish the subclasses from the superclasses. For example, to the right is the partial diagram of an ontology that has a partition of the *Car* class into the classes *2-Wheel Drive Car* and *4-Wheel Drive Car*. The partition rule (or subsumption rule) determines if a particular car is classified by the *2-Wheel Drive Car* or the *4-Wheel Drive Car* class.

If the partition rule(s) guarantee that a single *Car* cannot be in both classes, then the partition is called a disjoint partition. If the partition rules ensure that every concrete object in the super-class is an instance of at least one of the partition classes, then the partition is called an exhaustive partition.

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Ontology_components&action=edit&section=4)**] Attributes**

Objects in an ontology can be described by relating them to other things, typically aspects or parts. These related things are often called *attributes*, although they may be independent things. Each attribute can be a class or an individual. The kind of object and the kind of attribute determine the kind of relation between them. A relation between an object and an attribute express a fact that is specific to the object to which it is related. For example the [Ford Explorer](http://en.wikipedia.org/wiki/Ford_Explorer) object has attributes such as:

* <has as name> Ford Explorer
* <has by definition as part> *door* (with as minimum and maximum cardinality: 4)
* <has by definition as part one of> *{4.0L engine, 4.6L engine}*
* <has by definition as part> *6-speed transmission*

The value of an attribute can be a complex [data type](http://en.wikipedia.org/wiki/Data_type); in this example, the related engine can only be one of a list of subtypes of engines, not just a single thing.

Ontologies are only true ontologies if concepts are related to other concepts (the concepts do have attributes). If that is not the case, then you would have either a [taxonomy](http://en.wikipedia.org/wiki/Taxonomy) (if [hyponym](http://en.wikipedia.org/wiki/Hyponym) relationships exist between concepts) or a [controlled vocabulary](http://en.wikipedia.org/wiki/Controlled_vocabulary). These are useful, but are not considered true ontologies.

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Ontology_components&action=edit&section=5)**] Relationships**

Relationships (also known as relations) between objects in an ontology specify how objects are related to other objects. Typically a relation is of a particular type (or class) that specifies in what sense the object is related to the other object in the ontology. For example in the ontology that contains the concept Ford Explorer and the concept [Ford Bronco](http://en.wikipedia.org/wiki/Ford_Bronco) might be related by a relation of type <is defined as a successor of>. The full expression of that fact then becomes:

* Ford Explorer *is defined as a successor of* : Ford Bronco

This tells us that the Explorer is the model that replaced the Bronco. This example also illustrates that the relation has a direction of expression. The inverse expression expresses the same fact, but with a reverse phrase in natural language.

Much of the power of ontologies comes from the ability to describe relations. Together, the set of relations describes the [semantics](http://en.wikipedia.org/wiki/Semantics) of the domain. The set of used relation types (classes of relations) and their subsumption hierarchy describe the expression power of the language in which the ontology is expressed.



An important type of relation is the [subsumption](http://en.wikipedia.org/wiki/Subsumption) relation (*is-a-*[*superclass*](http://en.wikipedia.org/wiki/Superclass)*-of*, the converse of [*is-a*](http://en.wikipedia.org/wiki/Is-a), *is-a-subtype-of* or *is-a-*[*subclass*](http://en.wikipedia.org/wiki/Subclass)*-of*). This defines which objects are classified by which class. For example we have already seen that the class Ford Explorer *is-a-subclass-of* 4-Wheel Drive Car, which in turn *is-a-subclass-of* Car.

The addition of the is-a-subclass-of relationships creates a [taxonomy](http://en.wikipedia.org/wiki/Taxonomy); a tree-like structure (or, more generally, a [partially ordered set](http://en.wikipedia.org/wiki/Partially_ordered_set)) that clearly depicts how objects relate to one another. In such a structure, each object is the 'child' of a 'parent class' (Some languages restrict the is-a-subclass-of relationship to one parent for all nodes, but many do not).

Another common type of relations is the [mereology](http://en.wikipedia.org/wiki/Mereology) relation, written as *part-of*, that represents how objects combine together to form composite objects. For example, if we extended our example ontology to include concepts like Steering Wheel, we would say that a "Steering Wheel is-by-definition-a-part-of-a Ford Explorer" since a steering wheel is always one of the components of a Ford Explorer. If we introduce meronymy relationships to our ontology, the hierarchy that emerges is no longer able to be held in a simple tree-like structure since now members can appear under more than one parent or branch. Instead this new structure that emerges is known as a [directed acyclic graph](http://en.wikipedia.org/wiki/Directed_acyclic_graph).

As well as the standard is-a-subclass-of and is-by-definition-a-part-of-a relations, ontologies often include additional types of relations that further refine the semantics they model. Ontologies might distinguish between different categories of relation types. For example:

* relation types for relations between classes
* relation types for relations between individuals
* relation types for relations between an individual and a class
* relation types for relations between a single object and a collection
* relation types for relations between collections

Relation types are sometimes domain-specific and are then used to store specific kinds of facts or to answer particular types of questions. If the definitions of the relation types are included in an ontology, then the ontology defines its own ontology definition language. An example of an ontology that defines its own relation types and distinguishes between various categories of relation types is the [Gellish](http://en.wikipedia.org/wiki/Gellish) ontology.

For example in the domain of automobiles, we might need a *made-in* type relationship which tells us where each car is built. So the Ford Explorer is *made-in* [Louisville](http://en.wikipedia.org/wiki/Louisville%2C_Kentucky). The ontology may also know that Louisville is-located-in [Kentucky](http://en.wikipedia.org/wiki/Kentucky) and Kentucky is-classified-as-a state and is-a-part-of the [U.S.](http://en.wikipedia.org/wiki/United_States). Software using this ontology could now answer a question like "which cars are made in the U.S.?"

**SKOS Simple Knowledge Organization System Namespace Document - HTML Variant**

**18 August 2009 Recommendation Edition**

**This version:**

<http://www.w3.org/2009/08/skos-reference/skos.html>

**Latest version:**

<http://www.w3.org/2004/02/skos/core.html>

**Authors:**

[Alistair Miles](http://purl.org/net/aliman), STFC Rutherford Appleton Laboratory / University of Oxford
[Sean Bechhofer](http://www.cs.man.ac.uk/~seanb/), University of Manchester

**Status of this Document**

This document describes the schema available from the SKOS namespace.

**Introduction**

The Simple Knowledge Organization System (SKOS) is a common data model for sharing and linking knowledge organization systems via the Semantic Web.This document provides a brief description of the SKOS Vocabulary.

For detailed information about the SKOS Recommendation, please consult the SKOS Reference [[SKOS-REFERENCE](http://www.w3.org/2009/08/skos-reference/skos.html#SKOS-REFERENCE)] or the SKOS Primer [[SKOS-PRIMER](http://www.w3.org/2009/08/skos-reference/skos.html#SKOS-PRIMER)].

**SKOS Schema Overview**

The following table gives a non-normative overview of the SKOS vocabulary; it replicates a table found in the (normative) SKOS Reference [[SKOS-REFERENCE](http://www.w3.org/2009/08/skos-reference/skos.html#SKOS-REFERENCE)]. This document can be referenced directly, by its own URI <http://www.w3.org/2004/02/skos/core.html>, or indirectly, by content negotiation from the SKOS namespace URI [http://www.w3.org/2004/02/skos/core#](http://www.w3.org/2004/02/skos/core) as described in Appendix C of SKOS Reference [[SKOS-REFERENCE](http://www.w3.org/2009/08/skos-reference/skos.html#SKOS-REFERENCE)].

See also the SKOS Namespace Document - RDF/XML Variant [[SKOS-RDF](http://www.w3.org/2009/08/skos-reference/skos.html#SKOS-RDF)].

**Classes**

|  |
| --- |
| [**skos:Collection**](http://www.w3.org/2009/08/skos-reference/skos.html#Collection) |
| URI:  | http://www.w3.org/2004/02/skos/core#Collection  |
| Definition:  | [Section 9. Concept Collections](http://www.w3.org/TR/skos-reference/#collections) |
| Label:  | *Collection* |
| Disjoint classes:  | [skos:Concept](http://www.w3.org/2009/08/skos-reference/skos.html#Concept)[skos:ConceptScheme](http://www.w3.org/2009/08/skos-reference/skos.html#ConceptScheme) |
| [**skos:Concept**](http://www.w3.org/2009/08/skos-reference/skos.html#Concept) |
| URI:  | http://www.w3.org/2004/02/skos/core#Concept  |
| Definition:  | [Section 3. The skos:Concept Class](http://www.w3.org/TR/skos-reference/#concepts) |
| Label:  | *Concept* |
| Disjoint classes:  | [skos:Collection](http://www.w3.org/2009/08/skos-reference/skos.html#Collection)[skos:ConceptScheme](http://www.w3.org/2009/08/skos-reference/skos.html#ConceptScheme) |
| [**skos:ConceptScheme**](http://www.w3.org/2009/08/skos-reference/skos.html#ConceptScheme) |
| URI:  | http://www.w3.org/2004/02/skos/core#ConceptScheme  |
| Definition:  | [Section 4. Concept Schemes](http://www.w3.org/TR/skos-reference/#schemes) |
| Label:  | *Concept Scheme* |
| Disjoint classes:  | [skos:Collection](http://www.w3.org/2009/08/skos-reference/skos.html#Collection)[skos:Concept](http://www.w3.org/2009/08/skos-reference/skos.html#Concept) |
| [**skos:OrderedCollection**](http://www.w3.org/2009/08/skos-reference/skos.html#OrderedCollection) |
| URI:  | http://www.w3.org/2004/02/skos/core#OrderedCollection  |
| Definition:  | [Section 9. Concept Collections](http://www.w3.org/TR/skos-reference/#collections) |
| Label:  | *Ordered Collection* |
| Super-classes:  | [skos:Collection](http://www.w3.org/2009/08/skos-reference/skos.html#Collection) |

**Properties**

|  |
| --- |
| [**skos:altLabel**](http://www.w3.org/2009/08/skos-reference/skos.html#altLabel) |
| URI:  | http://www.w3.org/2004/02/skos/core#altLabel  |
| Definition:  | [Section 5. Lexical Labels](http://www.w3.org/TR/skos-reference/#labels) |
| Label:  | *alternative label* |
| Super-properties:  | http://www.w3.org/2000/01/rdf-schema#label |
| [**skos:broadMatch**](http://www.w3.org/2009/08/skos-reference/skos.html#broadMatch) |
| URI:  | http://www.w3.org/2004/02/skos/core#broadMatch  |
| Definition:  | [Section 10. Mapping Properties](http://www.w3.org/TR/skos-reference/#mapping) |
| Label:  | *has broader match* |
| Super-properties:  | [skos:broader](http://www.w3.org/2009/08/skos-reference/skos.html#broader)[skos:mappingRelation](http://www.w3.org/2009/08/skos-reference/skos.html#mappingRelation) |
| Inverse of: | [skos:narrowMatch](http://www.w3.org/2009/08/skos-reference/skos.html#narrowMatch) |
| [**skos:broader**](http://www.w3.org/2009/08/skos-reference/skos.html#broader) |
| URI:  | http://www.w3.org/2004/02/skos/core#broader  |
| Definition:  | [Section 8. Semantic Relations](http://www.w3.org/TR/skos-reference/#semantic-relations) |
| Label:  | *has broader* |
| Super-properties:  | [skos:broaderTransitive](http://www.w3.org/2009/08/skos-reference/skos.html#broaderTransitive) |
| Inverse of: | [skos:narrower](http://www.w3.org/2009/08/skos-reference/skos.html#narrower) |
| [**skos:broaderTransitive**](http://www.w3.org/2009/08/skos-reference/skos.html#broaderTransitive) |
| URI:  | http://www.w3.org/2004/02/skos/core#broaderTransitive  |
| Definition:  | [Section 8. Semantic Relations](http://www.w3.org/TR/skos-reference/#semantic-relations) |
| Label:  | *has broader transitive* |
| Super-properties:  | [skos:semanticRelation](http://www.w3.org/2009/08/skos-reference/skos.html#semanticRelation) |
| Inverse of: | [skos:narrowerTransitive](http://www.w3.org/2009/08/skos-reference/skos.html#narrowerTransitive) |
| Other characteristics: | Transitive |
| [**skos:changeNote**](http://www.w3.org/2009/08/skos-reference/skos.html#changeNote) |
| URI:  | http://www.w3.org/2004/02/skos/core#changeNote  |
| Definition:  | [Section 7. Documentation Properties](http://www.w3.org/TR/skos-reference/#notes) |
| Label:  | *change note* |
| Super-properties:  | [skos:note](http://www.w3.org/2009/08/skos-reference/skos.html#note) |
| [**skos:closeMatch**](http://www.w3.org/2009/08/skos-reference/skos.html#closeMatch) |
| URI:  | http://www.w3.org/2004/02/skos/core#closeMatch  |
| Definition:  | [Section 10. Mapping Properties](http://www.w3.org/TR/skos-reference/#mapping) |
| Label:  | *has close match* |
| Super-properties:  | [skos:mappingRelation](http://www.w3.org/2009/08/skos-reference/skos.html#mappingRelation) |
| Other characteristics: | Symmetric |
| [**skos:definition**](http://www.w3.org/2009/08/skos-reference/skos.html#definition) |
| URI:  | http://www.w3.org/2004/02/skos/core#definition  |
| Definition:  | [Section 7. Documentation Properties](http://www.w3.org/TR/skos-reference/#notes) |
| Label:  | *definition* |
| Super-properties:  | [skos:note](http://www.w3.org/2009/08/skos-reference/skos.html#note) |
| [**skos:editorialNote**](http://www.w3.org/2009/08/skos-reference/skos.html#editorialNote) |
| URI:  | http://www.w3.org/2004/02/skos/core#editorialNote  |
| Definition:  | [Section 7. Documentation Properties](http://www.w3.org/TR/skos-reference/#notes) |
| Label:  | *editorial note* |
| Super-properties:  | [skos:note](http://www.w3.org/2009/08/skos-reference/skos.html#note) |
| [**skos:exactMatch**](http://www.w3.org/2009/08/skos-reference/skos.html#exactMatch) |
| URI:  | http://www.w3.org/2004/02/skos/core#exactMatch  |
| Definition:  | [Section 10. Mapping Properties](http://www.w3.org/TR/skos-reference/#mapping) |
| Label:  | *has exact match* |
| Super-properties:  | [skos:closeMatch](http://www.w3.org/2009/08/skos-reference/skos.html#closeMatch) |
| Other characteristics: | TransitiveSymmetric |
| [**skos:example**](http://www.w3.org/2009/08/skos-reference/skos.html#example) |
| URI:  | http://www.w3.org/2004/02/skos/core#example  |
| Definition:  | [Section 7. Documentation Properties](http://www.w3.org/TR/skos-reference/#notes) |
| Label:  | *example* |
| Super-properties:  | [skos:note](http://www.w3.org/2009/08/skos-reference/skos.html#note) |
| [**skos:hasTopConcept**](http://www.w3.org/2009/08/skos-reference/skos.html#hasTopConcept) |
| URI:  | http://www.w3.org/2004/02/skos/core#hasTopConcept  |
| Definition:  | [Section 4. Concept Schemes](http://www.w3.org/TR/skos-reference/#schemes) |
| Label:  | *label* |
| Domain: | [skos:ConceptScheme](http://www.w3.org/2009/08/skos-reference/skos.html#ConceptScheme) |
| Range: | [skos:Concept](http://www.w3.org/2009/08/skos-reference/skos.html#Concept) |
| Inverse of: | [skos:topConceptOf](http://www.w3.org/2009/08/skos-reference/skos.html#topConceptOf) |
| [**skos:hiddenLabel**](http://www.w3.org/2009/08/skos-reference/skos.html#hiddenLabel) |
| URI:  | http://www.w3.org/2004/02/skos/core#hiddenLabel  |
| Definition:  | [Section 5. Lexical Labels](http://www.w3.org/TR/skos-reference/#labels) |
| Label:  | *hidden label* |
| Super-properties:  | http://www.w3.org/2000/01/rdf-schema#label |
| [**skos:historyNote**](http://www.w3.org/2009/08/skos-reference/skos.html#historyNote) |
| URI:  | http://www.w3.org/2004/02/skos/core#historyNote  |
| Definition:  | [Section 7. Documentation Properties](http://www.w3.org/TR/skos-reference/#notes) |
| Label:  | *history note* |
| Super-properties:  | [skos:note](http://www.w3.org/2009/08/skos-reference/skos.html#note) |
| [**skos:inScheme**](http://www.w3.org/2009/08/skos-reference/skos.html#inScheme) |
| URI:  | http://www.w3.org/2004/02/skos/core#inScheme  |
| Definition:  | [Section 4. Concept Schemes](http://www.w3.org/TR/skos-reference/#schemes) |
| Label:  | *is in scheme* |
| Range: | [skos:ConceptScheme](http://www.w3.org/2009/08/skos-reference/skos.html#ConceptScheme) |
| [**skos:mappingRelation**](http://www.w3.org/2009/08/skos-reference/skos.html#mappingRelation) |
| URI:  | http://www.w3.org/2004/02/skos/core#mappingRelation  |
| Definition:  | [Section 10. Mapping Properties](http://www.w3.org/TR/skos-reference/#mapping) |
| Label:  | *is in mapping relation with* |
| Super-properties: | [skos:semanticRelation](http://www.w3.org/2009/08/skos-reference/skos.html#semanticRelation) |
| [**skos:member**](http://www.w3.org/2009/08/skos-reference/skos.html#member) |
| URI:  | http://www.w3.org/2004/02/skos/core#member  |
| Definition:  | [Section 9. Concept Collections](http://www.w3.org/TR/skos-reference/#collections) |
| Label:  | *has member* |
| Domain: | [skos:Collection](http://www.w3.org/2009/08/skos-reference/skos.html#Collection) |
| Range: | union of [skos:Concept](http://www.w3.org/2009/08/skos-reference/skos.html#Concept) and [skos:Collection](http://www.w3.org/2009/08/skos-reference/skos.html#Collection) |
| [**skos:memberList**](http://www.w3.org/2009/08/skos-reference/skos.html#memberList) |
| URI:  | http://www.w3.org/2004/02/skos/core#memberList  |
| Definition:  | [Section 9. Concept Collections](http://www.w3.org/TR/skos-reference/#collections) |
| Label:  | *has member list* |
| Domain: | [skos:OrderedCollection](http://www.w3.org/2009/08/skos-reference/skos.html#OrderedCollection) |
| Range: | http://www.w3.org/1999/02/22-rdf-syntax-ns#List |
| Other characteristics: | Functional |
| [**skos:narrowMatch**](http://www.w3.org/2009/08/skos-reference/skos.html#narrowMatch) |
| URI:  | http://www.w3.org/2004/02/skos/core#narrowMatch  |
| Definition:  | [Section 10. Mapping Properties](http://www.w3.org/TR/skos-reference/#mapping) |
| Label:  | *has narrower match* |
| Super-properties:  | [skos:mappingRelation](http://www.w3.org/2009/08/skos-reference/skos.html#mappingRelation)[skos:narrower](http://www.w3.org/2009/08/skos-reference/skos.html#narrower) |
| Inverse of: | [skos:broadMatch](http://www.w3.org/2009/08/skos-reference/skos.html#broadMatch) |
| [**skos:narrower**](http://www.w3.org/2009/08/skos-reference/skos.html#narrower) |
| URI:  | http://www.w3.org/2004/02/skos/core#narrower  |
| Definition:  | [Section 8. Semantic Relations](http://www.w3.org/TR/skos-reference/#semantic-relations) |
| Label:  | *has narrower* |
| Super-properties:  | [skos:narrowerTransitive](http://www.w3.org/2009/08/skos-reference/skos.html#narrowerTransitive) |
| Inverse of: | [skos:broader](http://www.w3.org/2009/08/skos-reference/skos.html#broader) |
| [**skos:narrowerTransitive**](http://www.w3.org/2009/08/skos-reference/skos.html#narrowerTransitive) |
| URI:  | http://www.w3.org/2004/02/skos/core#narrowerTransitive  |
| Definition:  | [Section 8. Semantic Relations](http://www.w3.org/TR/skos-reference/#semantic-relations) |
| Label:  | *has narrower transitive* |
| Super-properties:  | [skos:semanticRelation](http://www.w3.org/2009/08/skos-reference/skos.html#semanticRelation) |
| Inverse of: | [skos:broaderTransitive](http://www.w3.org/2009/08/skos-reference/skos.html#broaderTransitive) |
| Other characteristics: | Transitive |
| [**skos:notation**](http://www.w3.org/2009/08/skos-reference/skos.html#notation) |
| URI:  | http://www.w3.org/2004/02/skos/core#notation  |
| Definition:  | [Section 6. Notations](http://www.w3.org/TR/skos-reference/#notations) |
| Label:  | *notation* |
| [**skos:note**](http://www.w3.org/2009/08/skos-reference/skos.html#note) |
| URI:  | http://www.w3.org/2004/02/skos/core#note  |
| Definition:  | [Section 7. Documentation Properties](http://www.w3.org/TR/skos-reference/#notes) |
| Label:  | *note* |
| [**skos:prefLabel**](http://www.w3.org/2009/08/skos-reference/skos.html#prefLabel) |
| URI:  | http://www.w3.org/2004/02/skos/core#prefLabel  |
| Definition:  | [Section 5. Lexical Labels](http://www.w3.org/TR/skos-reference/#labels) |
| Label:  | *preferred label* |
| Super-properties:  | http://www.w3.org/2000/01/rdf-schema#label |
| [**skos:related**](http://www.w3.org/2009/08/skos-reference/skos.html#related) |
| URI:  | http://www.w3.org/2004/02/skos/core#related  |
| Definition:  | [Section 8. Semantic Relations](http://www.w3.org/TR/skos-reference/#semantic-relations) |
| Label:  | *has related* |
| Super-properties:  | [skos:semanticRelation](http://www.w3.org/2009/08/skos-reference/skos.html#semanticRelation) |
| Other characteristics: | Symmetric |
| [**skos:relatedMatch**](http://www.w3.org/2009/08/skos-reference/skos.html#relatedMatch) |
| URI:  | http://www.w3.org/2004/02/skos/core#relatedMatch  |
| Definition:  | [Section 10. Mapping Properties](http://www.w3.org/TR/skos-reference/#mapping) |
| Label:  | *has related match* |
| Super-properties:  | [skos:mappingRelation](http://www.w3.org/2009/08/skos-reference/skos.html#mappingRelation)[skos:related](http://www.w3.org/2009/08/skos-reference/skos.html#related) |
| Other characteristics: | Symmetric |
| [**skos:scopeNote**](http://www.w3.org/2009/08/skos-reference/skos.html#scopeNote) |
| URI:  | http://www.w3.org/2004/02/skos/core#scopeNote  |
| Definition:  | [Section 7. Documentation Properties](http://www.w3.org/TR/skos-reference/#notes) |
| Label:  | *scope note* |
| Super-properties:  | [skos:note](http://www.w3.org/2009/08/skos-reference/skos.html#note) |
| [**skos:semanticRelation**](http://www.w3.org/2009/08/skos-reference/skos.html#semanticRelation) |
| URI:  | http://www.w3.org/2004/02/skos/core#semanticRelation  |
| Definition:  | [Section 8. Semantic Relations](http://www.w3.org/TR/skos-reference/#semantic-relations) |
| Label:  | *is in semantic relation with* |
| Domain: | [skos:Concept](http://www.w3.org/2009/08/skos-reference/skos.html#Concept) |
| Range: | [skos:Concept](http://www.w3.org/2009/08/skos-reference/skos.html#Concept) |
| [**skos:topConceptOf**](http://www.w3.org/2009/08/skos-reference/skos.html#topConceptOf) |
| URI:  | http://www.w3.org/2004/02/skos/core#topConceptOf  |
| Definition:  | [Section 4. Concept Schemes](http://www.w3.org/TR/skos-reference/#schemes) |
| Label:  | *is top concept in scheme* |
| Super-properties:  | [skos:inScheme](http://www.w3.org/2009/08/skos-reference/skos.html#inScheme) |
| Inverse of: | [skos:hasTopConcept](http://www.w3.org/2009/08/skos-reference/skos.html#hasTopConcept) |

**References**

**[SKOS-REFERENCE]**

[*SKOS Reference*](http://www.w3.org/TR/skos-reference), Alistair Miles, Sean Bechhofer, Editors. [Latest version](http://www.w3.org/TR/skos-reference) available at http://www.w3.org/TR/skos-reference.

**[SKOS-RDF]**

[*SKOS Namespace - RDF/XML Variant*](http://www.w3.org/TR/skos-reference/skos.rdf). [Latest version](http://www.w3.org/TR/skos-reference/skos.rdf) available at http://www.w3.org/TR/skos-reference/skos.rdf

**[SKOS-PRIMER]**

[*SKOS Primer*](http://www.w3.org/TR/skos-primer), Antoine Isaac, Ed Summers, Editors. [Latest version](http://www.w3.org/TR/skos-primer) available at http://www.w3.org/TR/skos-primer.

**Acknowledgements**

This document is the result of extensive discussions within the W3C's [Semantic Web Deployment Working Group](http://www.w3.org/2006/07/SWD/). The document drew on the experiences of earlier groups and projects including [SWAD-Europe](http://www.w3.org/2001/sw/Europe/) and W3C's [Semantic Web Best Practices and Deployment Working Group](http://www.w3.org/2001/sw/BestPractices/). Members of the W3C's [public-esw-thes](http://lists.w3.org/Archives/Public/public-esw-thes/) also made valuable contributions.