Asset Classification Taxonomy

# Introduction

* *Why, website, editors, internal control and management*
* *For whom*

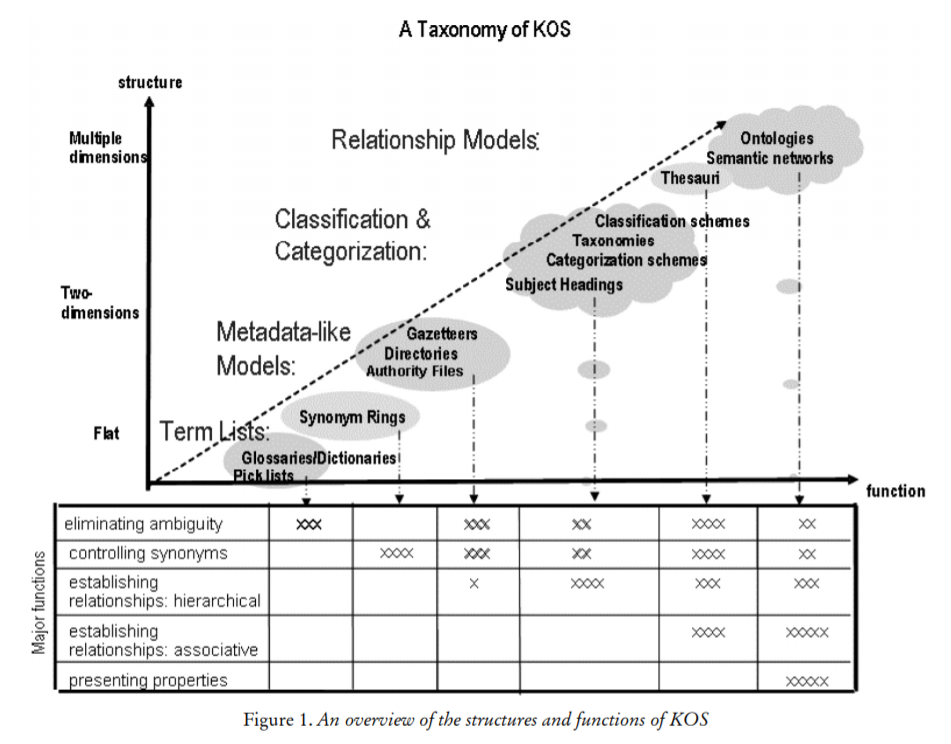
MDR publishes various reference assets (value vocabularies and models) ranging from simple value vocabularies, such as code lists, glossaries and taxonomies, to more complex models such as XML Schemas and cascading stylesheets, and up to semantically rich ontologies and application profiles. In order to facilitate and ensure proper authoring, documentation, publishing and management of these resources these assets need to be placed in well-defined categories. Moreover, a well-structured classification can enable users to easily find the desired content on the new EU-Vocabularies website.

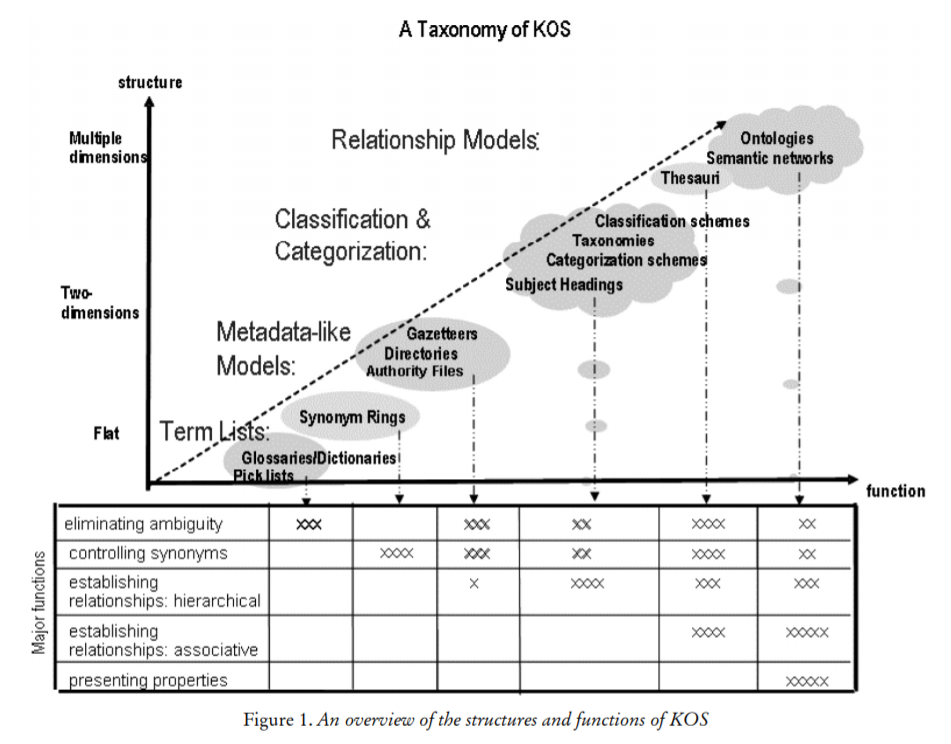
The Asset Classification taxonomy aims at classifying the reference meta-data resources. It constitutes a classification based on well-defined criteria and a glossary of terms with multiple definitions collected from various sources. This way the current classification with it's associated terminology can be used to establish cross-domain and cross organization links. For example, "coding structure" defined by UNSD *[ref.]* is the same as "classification scheme" defined in ANSI/NISO Z39 R2010 *[ref.]*. Or "faceted controlled vocabularies" defined by DLESE *[ref.]* is the same as synthetic classification defined by ODLIS *[ref.]*

*This document is structured as follows.*

# State of the art review

*Provide the list of references and their insights.*





# Methodology

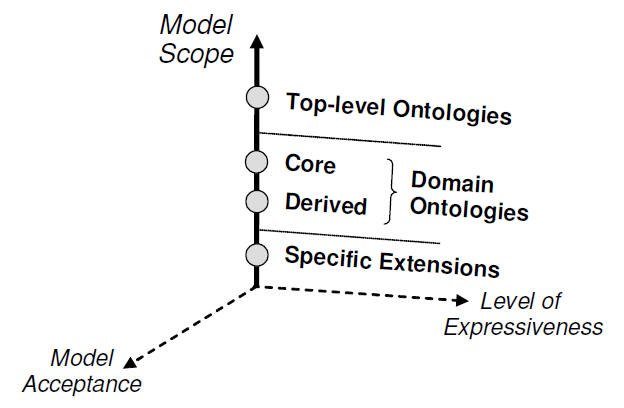
* Started from a list of definitions
* Sources: MDR table, Nist paper, other glossaries
* Investigated the properties of terms and definitions
* Choice of classification dimensions
  + Expressiveness, (Structure/function)
  + Acceptance scale
  + Scope scale

# Classification Dimensions

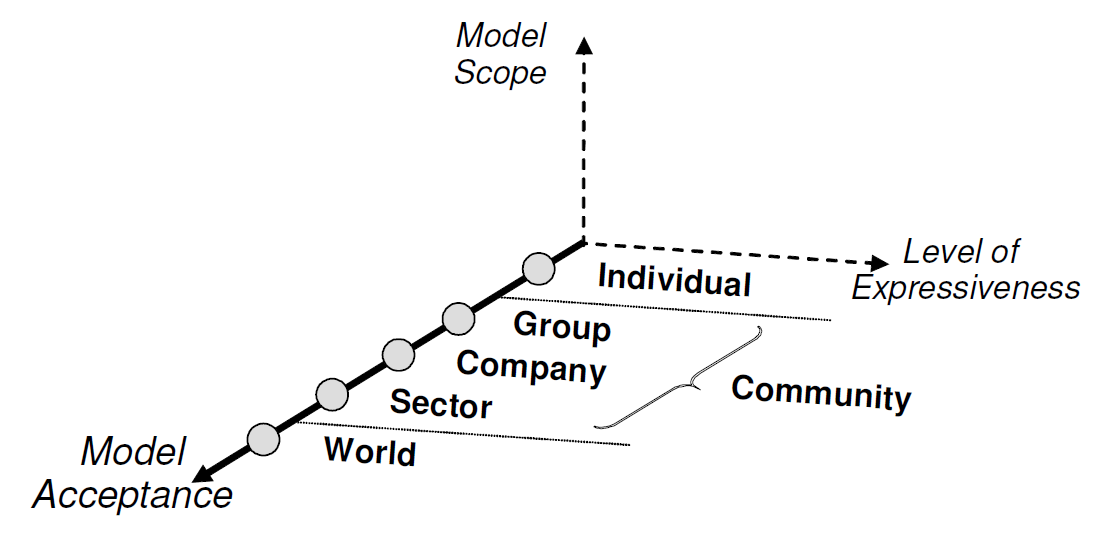
This section provides a description of important aspects when it comes to classifying reference assets

## Scope scale

"The model scope refers to the areas of semantics that is of interest to a model. It is important to understand that the scope does not restrict the observable individuals, but only the observable features of such individuals. Individuals with no features inside the model scope cannot be described and are therefore invisible for the model." [Schaffert2005]



## Acceptance scale

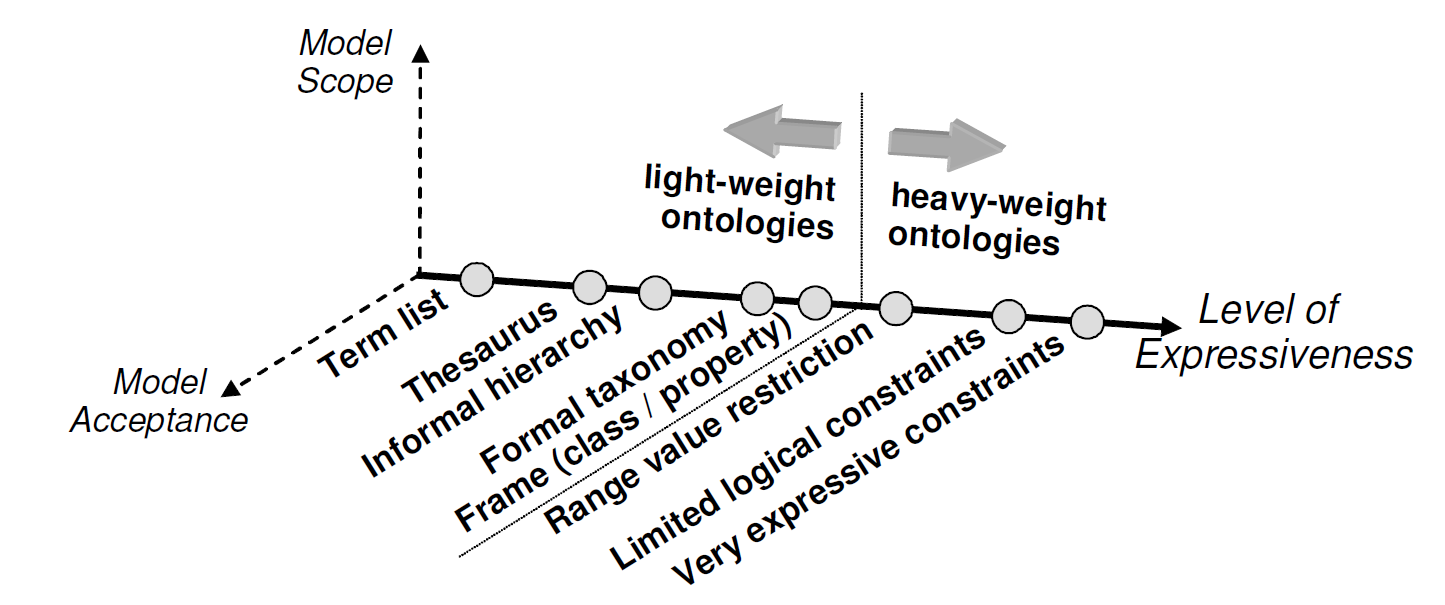


"The acceptance dimension is useful for classification of knowledge models w.r.t. two processes that are of importance for ontology engineering. The first aspect is to be clear about target communities of the application and its knowledge model. The second aspect deals with various methods of building consensus within a specific community. Involved communities include not only the intended users, but also the developers of a system. As far as web-based, distributed systems are concerned, wide acceptance is an important criterion of successful applications." [Schaffert2005]

## Expressiveness scale

### Corcho's expressiveness

Corcho distinguishes between the two main groups – light-weight ontologies and heavy-weight ontologies – and defines eight sub categories based on their level of expressiveness.



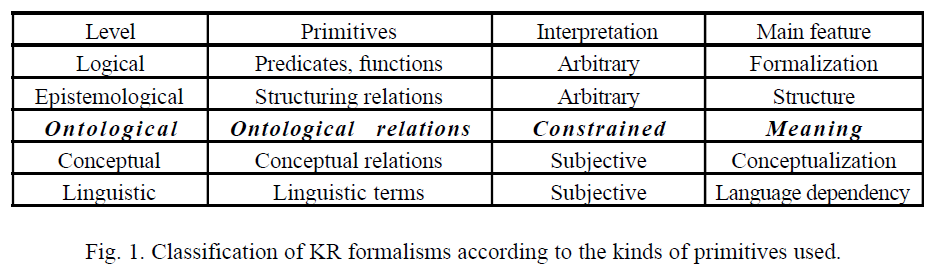
1. a term list or controlled vocabulary contains a list of keywords. Such lists are typically used to restrict possible values for properties of some kind of instance data in the domain;
2. a thesaurus also defines relations between terms, e.g. proximity of terms;
3. an informal taxonomy defines an explicit hierarchy (generalisation and specialisation), but there is no strict inheritance, i.e. an instance of a subclass is not necessarily also an instance of the superclass. Most available dictionaries like DMOZ are members of this category;
4. a formal taxonomy, in contrast, defines a strict inheritance hierarchy;
5. a frame or class/property based ontology is similar to object-oriented models. A class is defined by its position in the subclass hierarchy and its properties. Properties are inherited by subclasses and realised in instances;
6. a range value restriction defines, in addition, restrictions for the defined properties. Possible restrictions are data type or domain;
7. by using logic constraints, property values may be further restricted;
8. very expressive ontology languages often use first-order logic constraints. These constraints may include disjoint classes, disjoint coverings, inverse relationships, part-whole relationships, etc. [Schaffert2005]

### Harris's anatomy of an ontology

1. operational data
2. a knowledge model (or a part of such a model) can be used as reference data. Typically, there are no instances for this use case: the semantics defined in the model is only used to annotate instances built by another part of the model. This is e.g. common for term lists, thesauri, and taxonomies. Knowledge models used for such purposes are often referred to as “controlled vocabularies”.
3. a knowledge model can be used as a data structure. Here, the model is mainly used to work with operational data of the application, which means that instances are created based on the semantics defined in the model. Such knowledge models are often referred to as “data models” and are typically implemented by database management systems. The expressiveness needed for a knowledge model used as data structure is approximately the border between light- and heavy-weight ontologies. In practice, relational, objectoriented, and hierarchical models are used to define such models, but RDFS and OWL-DL are used more and more frequently.
4. a knowledge model can be used to **make assertions** or to **define constraints**. In this case, knowledge models and particular ontologies are used by the **application** to gain information about individuals by analysing constraints defined in the model e.g. by **integrity checking** and inferencing. This imposes the highest requirements on the expressiveness of the model. Models requiring these additional checks are usually referred to as domain models and are typically hard coded in the business logic of applications. The corresponding realisation, based on Semantic Web technologies, is to encode all semantics using a formal ontology language like first-order logic and to use a generic rule/inference engine to make the semantics operational.
5. Ontology definition language

[Schaffert2005]

### Guarino's ontological levels



[guarino1994-ontological levels]