

# Designing Ontology Schema and Data Instance for Nutrition Domain

Meena Unni<sup>1</sup> and Dr. K. Baskaran<sup>2</sup>

<sup>1</sup>Research Scholar, Department of Computer Science, Karpagam University  
Coimbatore 641 021, India

<sup>2</sup>Associate Professor, Department of Computer Science, Government College of Technology  
Coimbatore 641 013, India

## Abstract

The web has become a main research area. There is a large amount of information available on the web that is to be discovered, shared and utilized. Most of the information available on the web is unstructured and hence cannot be understood by machines. Semantic web is a step forward in this direction where data can be understood by both humans and machines. By encouraging the inclusion of semantic content in web pages, the Semantic Web aims at converting the current web dominated by unstructured and semi-structured documents into a "web of data". Ontology plays a critical role for the Semantic Web, and it is necessary to understand ontology in order to fully appreciate the idea of the Semantic Web. Ontology is defined as a common set of terms that are used to describe and represent a domain. It defines the terms used to describe and represent an area of knowledge. This paper explains in detail the development of ontology in the nutrition domain using Noy and McGuinness method.

**Keywords:** *Ontology, Semantic Web, Taxonomy, OWL, Class Hierarchy*

## 1. Introduction

Ontology is a formal explicit specification of a shared conceptualization [1]. Formal implies the fact that the ontology should be machine understandable. Explicit implies that the concepts and their constraints are unambiguously defined in the ontology. Shared implies that ontology should capture agreed knowledge. Conceptualization implies creating abstract model in the domain by precisely identifying relevant concepts

Ontology is domain specific, and it is used to describe and represent an area of knowledge. A domain is simply a specific subject area or area of knowledge, such as the area of photography, medicine, real estate, and education. Ontology contains terms and the relationships among these terms. Terms are often called classes, or concepts, and these words are interchangeable. The relationships

between these classes can be expressed by using a hierarchical structure: super classes represent higher level concepts and sub-classes represent finer concepts. The finer concepts have all the attributes and features that the higher concepts have. Ontology also represents properties which [2] describe various features and attributes of the concepts, and they can also be used to associate different classes together. Therefore, the relationships among classes are not only super class or sub-class relationships, but also relationships expressed in terms of properties.

Ontology comprises of ontology schema and instance. Ontology schema defines the terms and their relationships in the ontology whereas data instance focuses on augmenting the ontology with instances of concepts and properties. This part of the ontology can be kept in the knowledge base.

## 2. Designing Ontology

The methodology that is adopted in building the ontology is a modified version of the Agile Software Development proposed methodology called Methontology [3] which is a method to develop ontology from scratch.

Once all the information regarding vitamin A is collected, concepts of the ontology and their relationships were defined based on the collected information. Next step is to formalize the ontology by expressing concepts in a class subclass hierarchy. Then relationships between those classes and restrictions were added wherever applicable.

According to Noy and McGuinness [4], the following phases are to be followed to develop ontology.

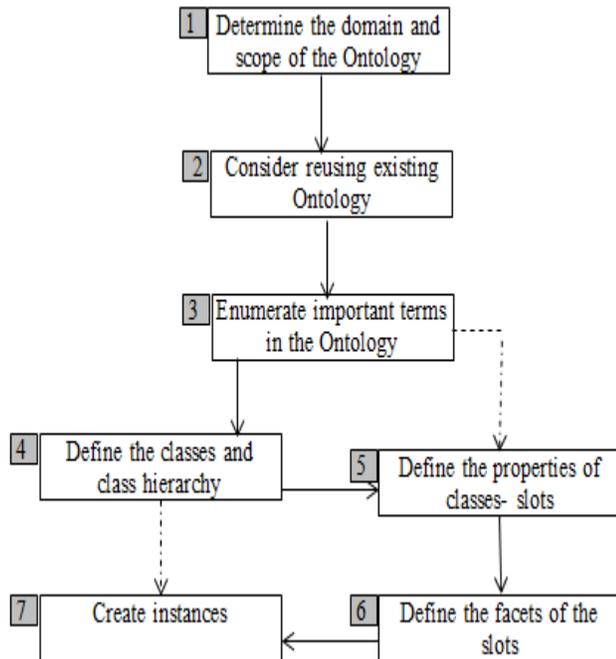


Fig. 1 Ontology development phases

### 2.1 Determine the domain and scope

Development of ontology starts with defining its scope and domain. There is need to study reasons for using the ontology. Considering the kinds of questions the ontology is supposed to answer is an important criteria while developing the ontology.

The ontology that is being developed for this work is that of Vitamin A. Representation of vitamin A for different dietary sources and its effect on human beings is the domain of the ontology. This ontology can be used to suggest diet with varying levels of vitamin A content. The concepts describing different types of diet both from plants and animals, effects of deficiency or excess intake of vitamin A on humans will be main focus of the ontology. Vitamin A requirement for different categories of humans, other diseases which may hinder the vitamin A effects too will figure into the ontology.

One of the ways to determine the scope of the ontology is to sketch a list of questions that a knowledge base based on the ontology should be able to answer, competency questions [5].

While the development of vitamin A ontology, the following competency questions have been considered.

What are the effects of deficiency of Vitamin A?

What is the recommended dietary allowance of vitamin A for different categories of people?

Which foods are rich in Vitamin A?

What are the different forms of Vitamin A?

How much vitamin A is there in each dietary source?

Does presence of other disease affect Vitamin A in the body?

Judging from this list of competency questions, the ontology will include information on various dietary sources, its effect on human, requirement amount of people, affect of diseases.

### 2.2 Consider reuse

There are ontologies available from third party that provides a starting point. If suitable ontology already exists, then it can be extended for the required domain. Research was carried out to find any existing ontologies on vitamin A, but could not locate any. Hence this ontology was developed from scratch.

### 2.3 Enumerate Key terms

It is useful to list all the terms that is likely to appear in the ontology. Classes, relationship between them and instances can be listed. For the chosen domain, important terms include Forms of Vitamin A, Effects on Human, Deficiency, Prevents, animal food, plant food, deficiency, overdose, person, Male, Female, Child, Interaction, Tolerable Upper limit, Causes, Enhances, May Result, Suffers from etc. Fundamentally, it is useful to obtain a comprehensive list of terms without worrying the overlap between the term and concept of the property, or concept includes slot or class.

Developing the class hierarchy and defining properties of concepts (slots) are closely interdependent. These two steps are also the most important steps in the ontology design process.

### 2.4 Define taxonomy

Once the key terms are identified, they have to be organised into a superclass-subclass hierarchy, which is also known as taxonomy. Subclasses specialise their super classes. There are several approaches in the development of class hierarchies [6].

Class hierarchy can be developed using top-down, bottom-up or combinational approach.

- Top-down development starts with defining the most general concepts in the domain and then deriving more specific concepts under it. In order to create ontology on Vitamin A, general concepts like diet, person, and effects can be identified. Then subclasses can be created for these concepts. For example *DietarySource* can be further categorised as *AnimalSource* and *PlantSource*. This work uses top down developmental approach.
- Bottom-up development starts with defining the specific classes, the leaves of the hierarchy and then all these classes can be grouped to form more general concept. For example classes *Fruit*, *Vegetable*, *Lentil* can be created. Then a common superclass *PlantSource* for these two classes is created, which in turn is a subclass of *DietarySource*.
- Combination development combines both topdown and bottom-up approach. Some classes are created using top down approach and others are created using bottom up approach.

Person	Infant1-3		
	Child	Between4-8 Between9-13	
	Female	Lactating Pregnant Women	
	Male		
Interaction	PositiveInteraction NegativeInteraction		
FormsOf VitaminA	PreformedVitaminA ProvitaminACareten oid		
Supplement			

Table 1. The different levels of the Vitamin A taxonomy represented in a tabular form

Table 1 shows top level, level1, level2 level 3 and level of the proposed ontology in the nutrition domain

Top	Level 1	Level 2	Level 3	Level 4
DietarySource	AnimalSource	Dairy	Butter Cheese Curd Milk	
			Egg Fish Meat	
EffectOnHuman	PlantSource	Fruit Lentil Vegetable NutAndSeed WholeGrain		
			Deficiency	EyeDisease SkinDisease OtherDisease
	ExcessIntake WithinLimit			

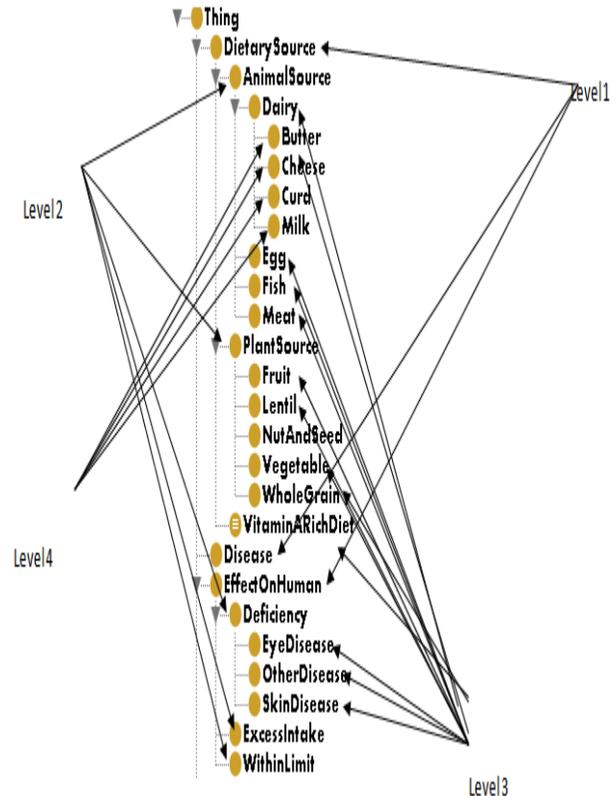


Fig. 2 Different levels of taxonomy represented using ontology editor

Fig 2 represents different levels of taxonomy using ontology editor. Ontology editor is a useful tool to represent information in the class subclass hierarchical structure.

Vitamin A ontology is designed which is used by every aspect of the system, in the information extraction, inference and retrieval phases. An iterative development process in the ontology engineering phase is followed till a stable ontology containing nearly classes and appropriate properties are obtained.

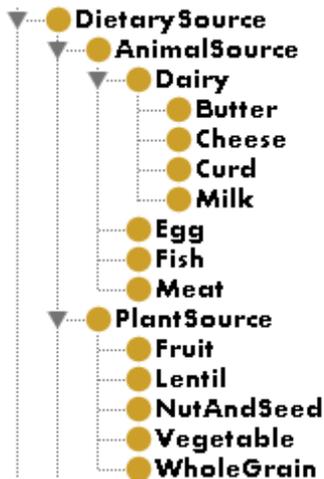


Fig. 3 Figure showing class and subclass under DietarySource

Figure shows class subclass hierarchy for *DietarySource*. This view shows two types of class, primitive and defined class. *VitaminARichDiet* is a defined class; all other classes are primitive classes. A class that only has necessary conditions is known as a Primitive Class. A class that has at least one set of necessary and sufficient conditions is known as a Defined Class.

### 2.5 Define the properties of the classes

A property defines relationship between object. If the relationship is between two individuals, it is an object property. Data property defines relationship between an individual and data values. *Enhances*, *IsEnhancedby*, *Causes* are examples of object properties. *Causes* defines relationship between *Dietarysource* and *ExcessIntake*. Whereas *HasQuantity*, *HasVitaminAinIU* are examples of data property. *HasVitaminAinIU* gives the quantity of vitamin A in an instance say plant food carrot.

For Example

```
<owl:ObjectProperty rdf:about="#Causes">
```

```
<rdfs:domain rdf:resource="#DietarySource"/>
<rdfs:rangerdf:resource="#ExcessIntake"/>
<owl:inverseOf rdf:resource="#IsCausedBy"/>
</owl:ObjectProperty>
```

The above code defines an object property *Causes* has *DietarySource* has its domain, and has *ExcessIntake* as range. Inverse property of *Causes* is *IsCausedBy*. If one attribute has many domains then its domain will be intersection of its entire domain [7].

The classes, to which a property is attached, are called the domain of the slot. Allowed classes for properties of type Instance are often called a range of a property.

```
<owl:ObjectProperty rdf:about="#SuffersFrom">
  <rdf:type
  rdf:resource="&owl:FunctionalProperty"/>
  <rdfs:comment rdf:datatype="&xsd:string"
  >This slot indicates the disease suffered
  by a particular person</rdfs:comment>
  <rdfs:range rdf:resource="#Disease"/>
  <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>
```

The above code defines an object property *SuffersFrom* which has *Person* has its domain, and *Disease* as range. Object property *SuffersFrom* defines relationship between the class *Person* and *Disease*.

The OWL code given below explains how an instance *John* who is a member of the class *Male* is related to *CysticFibrosis* which is a member of the class *Disease*.

```
<owl:Thing rdf:about="#John">
  <rdf:type rdf:resource="#Male"/>
  <SuffersFrom rdf:resource="&www;CysticFibrosis"/>
</owl:Thing>
```

For each property, there is need to determine which class it describes. These properties become slots attached to classes. Thus, some of the slots of *vegetable* class are *hasVegType*, *HasLycopene*, *HasName*, *HasQuantity* etc. All subclasses of a class inherit the slot of that class. All the slots of the class *person* will be inherited by all its subclasses like *Women*, *Male*, *Child*.

### 2.6 Define the facets of a class

A facet of a property explains the value type, allowed values, the cardinality and other features of the values the property can take (Noy and McGuinness, 2000).

For example consider the property *HasVitaminAinIU*. The value type of this is 'Float' and cardinality is one. That is *HasVitaminAinIU* is a slot with value type Float. A slot can have multiple values and the values are instances of the class.

Slot cardinality: Slot cardinality defines how many values a slot can have. Single cardinality allows only one value for the slot whereas multiple cardinality allows more than one value for the slot. *HasVitaminAinIU* is a single cardinality slot. Slot value type describes what types of values can fill in the slot. There are number of value types like Boolean, integer, float, string, XML, Literal, URI to name a few. *HasVitaminAinIU* has value type as Float where as *HasName* has value type as string.

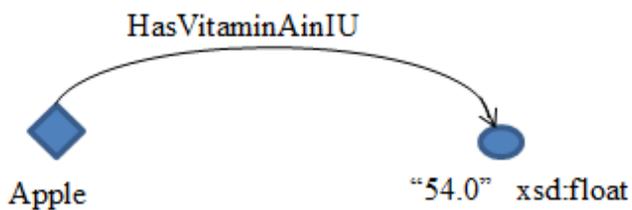


Fig. 1 Datatype property linking the individual Apple to the data literal '54.0.', which is of type float

The same information is represented in OWL code as shown below.

```
<Fruit rdf:about="#Apple">
<HasVitaminAinIU
rdf:datatype="&xsd;float">54.0</HasVitaminAinIU>
</Fruit>
```

### 2.7 Define instances

This is the last step in creating ontology. Instances can be manually typed or retrieved from another source like database or web page. Defining an individual instance of a class requires first choosing a class, secondly creating an individual instance of that class, and finally filling in the slot values.

To create an individual instance of *Cabbage*, the class under which it has to be created should be determined. Later slot values can be filled. Since *Cabbage* is a type of

*Vegetable* which in turn is a *PlantSource* which is under *DietarySource*. Different slot values are as follows.

HasAlphaCarotene	33.0
HasName	Cabbage
HasQuantity	100
HasLuteinZeaxanthin	30.0
HasUnitOfMeasurement	gram
HasLycopene	0.0
HasVitaminAinIU	98.0
HasPlantType	Annual
HasBetaCarotene	42.0
HasVegType	Leaf
HasRetinolActivityEquivalent	5.0
HasBetaCryptoxanthin	0.0
HasColor	Green
HasMainConstituent	Cabbage

The OWL code for representing the attributes of cabbage is given below.

```
<owl:Thing rdf:about="#Cabbage">
  <rdf:type rdf:resource="#Vegetable"/>
  <HasLycopene
rdf:datatype="&xsd;float">0.0</HasLycopene>
  <HasBetaCryptoxanthin
rdf:datatype="&xsd;float">0.0</HasBetaCryptoxanthin>
  <HasQuantity
rdf:datatype="&xsd;int">100</HasQuantity>
  <HasLuteinZeaxanthin
rdf:datatype="&xsd;float">30.0</HasLuteinZeaxanthin>
  <HasAlphaCarotene
rdf:datatype="&xsd;float">33.0</HasAlphaCarotene>
  <HasBetaCarotene
rdf:datatype="&xsd;float">42.0</HasBetaCarotene>
  <HasRetinolActivityEquivalent
rdf:datatype="&xsd;float">5.0</HasRetinolActivityEquivalent>
  <HasVitaminAinIU
rdf:datatype="&xsd;float">98.0</HasVitaminAinIU>
  <HasPlantType
rdf:datatype="&xsd;string">Annual</HasPlantType>
  <HasMainConstituent
rdf:datatype="&xsd;string">Cabbage</HasMainConstituent>
  <HasName
rdf:datatype="&xsd;string">Cabbage</HasName>
  <HasColor
rdf:datatype="&xsd;string">Green</HasColor>
  <HasVegType
rdf:datatype="&xsd;string">Leaf</HasVegType>
```

```
<HasUnitOfMeasurement  
rdf:datatype="&xsd:string">gram</HasUnitOfMeasurem  
ent>  
</owl:Thing>
```

### 3. Conclusion

This paper describes designing ontology using Noy and McGuinness steps. Ontology on Vitamin A is created and it is instantiated with data values. Determining the domain and scope of the ontology is the foremost step while developing ontology. Defining taxonomy by identifying the key terms is the next important step. Representing the properties and facets of the class culminates the creation of ontology schema. Once the schema is created, instances are added to the ontology. Queries can be performed over the ontology to retrieve instances satisfying the given condition. This also results in validating the rules and logics of the ontology.

### References

- [1] Dieter Fensel · HolgerLausen · Axel Polleres Jos de Bruijn · Michael Stollberg · Dumitru Roman John Domingue, Enabling Semantic Web Services, springer, 2007
- [2] Liyang Yu, 2011, A developers guide to the semantic web, Springer-Verlag Berlin Heidelberg,
- [3]M Fernandez-Lopez, A Gomez-Perez, N Juristo.1997.Methontology: from ontological art towardsontological engineering, Spring Symposium on Ontological Engineering of AAAI. Stanford University, California. 1997.
- [4] Noy, N., McGuinness, D.L, Ontology Development 101: A guide to creating your first ontology, Stanford Medical Informatics Technical Report No SMI-2001-0880, 2000
- [5] Gruninger, M. and Fox, M.S. (1995). Methodology for the Design and Evaluation of Ontologies. In: Proceedings of the Workshop on Basic Ontological Issues in KnowledgeSharing, IJCAI-95, Montreal
- [6] “DL Query Tab”, p. 1, <http://protegewiki.stanford.edu/index.php/DLQueryTab> . Accessed on May 20th 2009.
- [7] Heming Gong, Jianyi Guo —Research and Building and Reason-ing of Travel Ontologyl IEEE-2008.