Contaduría y administración
ISSN: 0186-1042
ISSN: 2448-8410
Facultad de Contaduría y Administración, UNAM

Bravo, Maricela; Hoyos Reyes, Luis Fernando; Reyes Ortiz, José A.
Methodology for ontology design and construction
Contaduría y administración, vol. 64, no. 4, e134, 2019, October-December
Facultad de Contaduría y Administración, UNAM

DOI: https://doi.org/10.22201/fca.24488410e.2020.2368

www.cya.unam.mx/index.php/cya

Contaduría y Administración 64 (4), 2019, 1-24

Contaduría y Administración
Accounting & Management

# Methodology for ontology design and construction

*Metodología para el diseño y construcción de la ontología*

Maricela Bravo*, Luis Fernando Hoyos Reyes, José A. Reyes Ortiz
Universidad Autónoma Metropolitana, México

Received January 9, 2019; accepted March 15, 2019
Available online March 22, 2019

## Abstract

In this paper we present a methodology for ontology design and construction which incorporates the most outstanding design principles and a thorough evaluation process. An ontology provides logical formulation of complex problems of decision sciences like risk management, decision making under uncertainty, statistics and forecasting, negotiation and financial analysis. The main stages of this methodology are: requirements specification, formal design, construction, and evaluation. At each stage of the methodology a series of tasks are defined together with methods and techniques to build the ontology considering quality characteristics. Description logics is used as the formal language during design, a set of informal competency questions is used to support ontology conceptualization; at the evaluation stage, the set of questions are translated to a formal reasoning language and are used for evaluation purposes. Many methodologies and tools have been reported in literature, but little attention has been paid in the creation of consistent, modular, coherent, usable and reusable ontologies as an objective from the beginning of the design process. A comparative analysis with other methodologies is discussed and an Ontological Model for Medical Diagnosis is presented.

*JEL code:* C63, D80, I10, L86
*Keywords:* Ontology design; Ontology design principles; Ontology evaluation; Decision sciences

*Corresponding author.
E-mail address: mcbc@azc.uam.mx (Maricela Bravo)
Peer Review under the responsibility of Universidad Nacional Autónoma de México.

http://dx.doi.org/10.22201/fca.24488410e.2020.2368

http://dx.doi.org/10.22201/fca.24488410e.2020.2368

**Resumen**

En este artículo se presenta una metodología para el diseño y la construcción de ontologías que incorpora los principios de diseño más destacados y un proceso de evaluación exhaustivo. Una ontología proporciona una formulación lógica para la representación de problemas complejos en ciencias de la decisión, como la gestión de riesgos, la toma de decisiones en situaciones de incertidumbre, estadísticas y pronósticos, negociación y análisis financiero. Las etapas principales de esta metodología son: especificación de requisitos, diseño formal, construcción y evaluación. En cada etapa de la metodología se definen una serie de tareas junto con métodos y técnicas para construir la ontología considerando las características de calidad. Las lógicas descriptivas se utilizan como lenguaje formal durante el diseño, un conjunto de preguntas de competencia informal se usa para apoyar la conceptualización de la ontología; en la etapa de evaluación, el conjunto de preguntas se traduce a un lenguaje de razonamiento formal y se utiliza para fines de evaluación. Diversas metodologías y herramientas se han reportado en la literatura, pero se ha prestado poca atención en la creación de ontologías consistentes, modulares, coherentes, utilizables y reutilizables como objetivo desde el inicio del proceso de diseño. Se discute un análisis comparativo con otras metodologías y se presenta un Modelo Ontológico para el Diagnóstico Médico.

*Código JEL:* C63, D80, I10, L86

*Palabras clave:* Diseño de ontologías; Principios de diseño de ontologías; Evaluación de ontologías; Ciencias de la decisión.

## Introduction

For more than two decades the term ontology has been acquiring greater interest among researchers, academics and professionals from different areas of knowledge who have seen the need to use, design or apply ontologies as a solution mechanism to their information systems requirements. The term ontology whose origin is found in Philosophy was adopted by the research community of Artificial Intelligence to formally describe domains of knowledge. Many definitions of the term ontology have been proposed, for example Neches et al. (1991) stated that an ontology "*defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary*". Latter Gruber (1993) defined "*An ontology is an explicit specification of a conceptualization*", in other words, an ontology is a model of a domain (conceptualization) that is explicitly described (in the form of a specification). Lassila and McGuinness (2001) defined the mandatory properties for an explicit specification of a conceptualization to be considered an ontology: finite and controlled vocabulary, not ambiguous interpretation of classes and relations between terms, and strict hierarchies of the relations of subclasses between classes.

In summary, to facilitate the understanding of the term ontology two descriptions are utilized: from the constitutive elements of ontologies, and from the characteristics that an ontology must present.

a) An ontology defines a set of concepts or classes (vocabulary or conceptualization), the taxonomic relations (hierarchical) between those concepts, the semantic relationships between concepts, individuals or instances, and axioms.
b) Likewise, an ontology must comply with the characteristics of being a formal (expressed in a formal language), explicit (concepts should be made explicit through axioms) and a represent a shared specification (based on the consensus of a group of experts).

Another important aspect of ontologies is the possibility of executing reasoning and inference to produce new concepts or semantic relationships, this is possible due to the formal logic languages that underlie the representation of ontologies. Therefore, an additional element of ontologies is the set of inference rules for knowledge and information gathering, and the set of axioms that produce new definitions through reasoning engines.

Currently there are many professionals related to engineering who have faced the need to build a knowledge base based on ontologies without having a good methodological guide to support them. In this way, many engineers and solution integrators have faced the need for an easy-to-implement methodology that allows them to build quality models that solve very specific industry problems.

The construction of ontologies covers methods, techniques and design principles that have been proposed to support the efficient design of ontologies. The implementation of a methodology for efficient ontology construction aims at producing ontologies that are usable, reusable and easy to maintain. Gómez-Pérez (1999) clarified that a methodology is composed of methods, techniques, processes and activities. Accordingly, in this work the methodology for ontology design and construction is defined as an ordered series of phases that specify the procedures used in the engineering of an ontology or ontology system.

The rest of the paper is organized as follows: the next section describes the proposed methodology and its phases; the third section the application of the methodology for medical diagnosis support is presented as a case of study; comparative analysis of a collection of related methodologies is discussed in fourth section. The final section summarizes the main conclusions.

## Proposed Methodology

In this section, a comprehensive ontology design and construction methodology is described, its main goal is to assist ontology developers by providing methods and techniques to support ontology construction adhering to design principles. For the purpose of this work, an ontology system is a global ontology that imports individual ontologies which are semantically related

between them inside the global ontology. Figure 1 shows the phases and procedures of the proposed methodology.

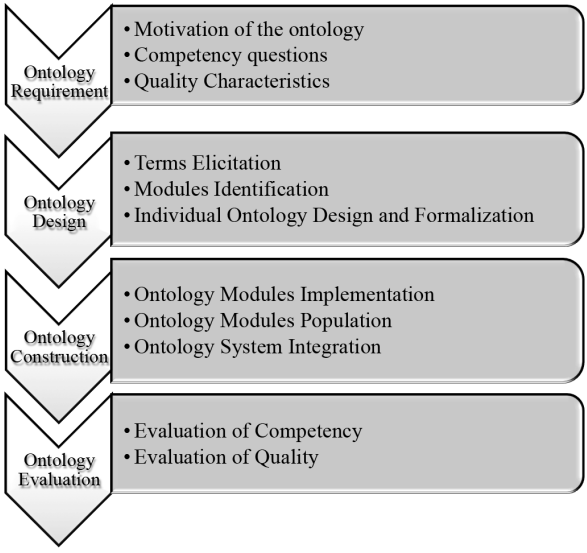


Figure 1. Phases and procedures of the proposed methodology.

Source: Author's own.

The main characteristics of the proposed methodology are:

a) Modular oriented. Modularization of ontologies is a key factor for achieving reutilization and maintainability. Modules represent domain spaces identified form the universe of concepts.
b) Domain-oriented. The hierarchical organization of concepts is done by grouping all domain-related terms and assigning a class identification.
c) User-centered design and evaluation. This methodology considers the specific requirements of the group of users as they are involved in the definition of the competency questions from the beginning of the methodology to the evaluation.
d) Incremental and iterative. The proposed methodology is defined as a series of phases, but inner in the specific procedures an incremental construction is defined.
e) Quality-oriented. Quality requirements are proposed at the beginning of the design process, these quality requirements are used also at the final phase of evaluation, to verify accomplishment.

This proposed methodology is defined to work with teams integrated with a group of knowledge domain experts, a group of computer programmers and analysts with experience implementing ontologies and applications that exploit ontologies. And one or two ontology engineers.

**Ontology Requirements Specification**

The objective of the specification of requirements is the identification of the scope of the ontology, the definition of possible scenarios, users, the competence of the ontology, and the quality characteristics that it must attend. In order to specify the requirements of the ontology, the following tasks must be executed:

a) Specify the motivation of the ontology, clarifying the possible scenarios, users and applications that will benefit.
b) Specify the competency of the ontology by consensus within a group of experts in the domain of the ontology and ask them to generate a list of competence questions, that is, questions that they want the ontology to answer. The ontology engineer together with the group of domain experts produce a list of competency questions. Such competency questions are generated by asking the group of domain experts to enunciate direct questions that they expect the ontology system will be able to answer once it is implemented and in production. The list of competency questions will also be useful for the final ontology evaluation.

**Ontology Design**

The *second phase* of the methodology aims at producing a formal design of the ontology. This phase consists of the following procedures: term elicitation, ontology modules identification, individual ontology design and formalization using Description Logics (DL) notation. DL´s are formal languages designed for knowledge representation and reasoning; DL´s are decidable fragments of First Order Logic (FOL).

a) Term elicitation consists of producing a seminal list of terms that are relevant for the particular domain of knowledge. In order to achieve this goal, the following activities are defined: using the list of competency questions, the ontology engineer identifies the elementary concepts (nouns) that are required for the representation of the ontology model.
b) Modules identification consists of deciding the set of individual ontologies that will conform the ontology system. In order to produce this set of ontology modules, it is necessary to group similar terms by using as input the list of terms, the ontology engineer together with the group of domain expert's workout the clustering of terms.

This clustering consists of putting together all domain-related terms and assigning a class identification with its name in singular. The resulting list of classes represents the set of ontology modules. Each group is then converted into a single ontology, aiming at separating groups of different domains allowing that each individual ontology can be reused for other applications. As a result of this phase a global scheme of the set of individual ontologies that will integrate the system is obtained.

c) Individual ontology design and formalization consists of defining the elementary terminological axioms of each individual ontology using DL. The following activities are necessary to achieve this design:

*I.* *Hierarchical relationships* definition. For each group of terms that belong to the same ontology, hierarchical relations should be defined, adding more concepts into the hierarchy if necessary.

*II.* *Data properties relationships* definition. Data properties represent the data attributes of each concept. For instance, a Person has name, has age, has weight, etc.

*III.* *Object properties relationships* definition. Object property relations are those relations that link semantically concepts into an ontology, the difference between object properties and data type properties is that object properties relate two concepts while data properties relate a concept with a data type. In order to define these relations into individual ontologies, the knowledge engineer should answer the question: Is there any semantic relation between concepts of this ontology?

*IV.* *Axiomatization* consists of defining the minimal set of characteristics that every individual should fulfil to be a member of a class or concept. The following types of axioms can be defined during this activity: class axioms, cover axioms, closure axioms, data and object property axioms allowing the detailed specification of: cardinality restrictions, existential and universal restrictions, and value restrictions.

## Ontology Construction

The objective of the *third phase* of the methodology is to code all ontology modules by using an ontology editor and a standard language, and integrate all modules into an ontology system. For each ontology module the following activities are defined:

a) Implementation consists of utilizing an editor to implement the ontology modules defined during the previous phase. Although there are various ontology editors available, our recommendation is to use Protégé[1], an open-source ontology editor that is kept up to date and with broad support from the developer community.

b) Modules population consists of instantiating the ontology with individuals to evaluate the initial definitions, relations and axioms. For each individual ontology consistency checking is executed to verify that none of the class definitions and axioms has logical

[1] https://protege.stanford.edu/

contradictions. This final activity consists of executing the reasoning tasks of taxonomy classification, compute inferred types, and consistency checking.

c) Modules integration consists of importing the set of individual ontologies into the ontology system. Once all individual ontologies are imported into the ontology system a consistency checking is mandatory to verify that the integration of individual ontologies do not produce any contradiction. To define the semantic relations between concepts in the integrated ontology, the following activities are done:

*I*. *Ontology system properties* definitions. Using the initial list of competency questions and the list of verbs, the ontology engineer should identify and create all necessary object properties between concepts from the different ontologies.

*II*. *Ontology system axiomatization*. Once the integrated ontology system is available, as well as the properties between objects, the creation of axioms and specific restrictions is carried out.

*III*. *Ontology system population*. Finally, the ontology system is filled with individuals that require semantic relationships between ontologies. This final step is used also for evaluation purposes, correction of object properties and axioms is done if necessary during this step.

## Ontology Evaluation

According with Gómez-Pérez (1994), Ontology Evaluation refers to the correct building of the content of ontology, ensuring that its definitions correctly implement the ontology requirements and competency questions. The goal of ontology evaluation is to prove compliance of the world model with the world modelled formally. "Evaluation means to judge the ontologies, their associated software environments and documentation with respect to a frame of reference during each phase and between phases of their life cycle". Two important aspects are used for evaluation: the competency of the ontology and the quality requirements.

a. Competency of the Ontology. Gruninger and Fox (1994) proposed six characteristics to evaluate a Business Model. These characteristics were proposed to answer the question of "How can one determine which model is correct for a given task?" To give a guideline on the operation of these characteristics, the authors define the concept of competence of the model as follows: "*given an appropriately instantiated model and a demonstrator of theorems, the competence of a model is the set of questions that the model should be able to answer*". Based on this definition, it is possible to state that the competence of an ontology model is the set of questions that the ontology is capable of answering. Evaluation of the competency of an ontology system is crucial to verify that a representational model is complete with respect to a given set of competency questions. Evaluation of the competency of the ontology requires the translation of all

competency questions into Description Logic (DL) axioms to assure that the ontology system can answer the initial set of competency questions.

b. Quality Requirements. Over the years many researchers have presented and discussed ontology design principles as objective criteria which represent guidelines for the design and evaluation of ontology models, Gruber (1993). Therefore, the quality of an ontology model can be measured as the degree of compliance it has with respect to established design criteria. There are many ontology design principles reported in specialized literature, among others, the following design principles have been selected for the design and construction of the ontology model reported in this paper:

*I*. *Clarity*. According with Gruber (1993), the ontology should communicate the intended meaning of defined terms. Definitions should be stated in formal axioms, and a complete definition (defined by necessary and sufficient conditions) is preferred over a partial definition (defined by only necessary or sufficient conditions). However, the incorporation of complete definitions has a very high cost in relation to memory usage and computing resources. Therefore, this criterion should be carefully considered due to the trade-off between clarity and performance.

*II*. *Coherence*, this principle is also referred as soundness or consistency. Coherence specifies that ontology definitions should be individually sound and should not contradict each other. Accordingly, the coherence of an ontology can be verified by executing the consistency checking of any reasoner program.

*III*. *Modularity*, modularization of ontologies is defined as the task of decomposing an ontology into independent disjoint skeleton taxonomies. Regarding modularization of ontologies, Rector (2003) stated the importance of modularity as a key requirement for ontologies in order to achieve reutilization, maintainability, and evolution.

Quality evaluation is done by consistency checking to verify that none of the class definitions and axioms has logical contradictions, nor the individuals' instantiated into the ontology. This final activity consists of executing the reasoning tasks of taxonomy classification, compute inferred types, and consistency checking.

## Case of Study: Ontological Model for Medical Diagnosis

In this section we describe a case study of the application of the proposed methodology. The objective is to design and implement an ontology system to support the decision making during the diagnosis of medical diseases.

http://dx.doi.org/10.22201/fca.24488410e.2020.2368

**Ontology Requirements Specification**

After a series of interviews with a group of doctors (physicians), programmers and a main knowledge engineer, the following requirements were specified:

a. This ontology model aims at supporting the general medic (Physician) to review of signs and symptoms of the patient with some disease, and to support the decision making regarding the additional laboratory tests to determine the diagnosis and adequate treatment considering the particular characteristics and profile of the patient.
b. As a result of the meetings with the group of experts, the following list of competence questions was defined:
    - ✓ *What are the symptoms of a disease?*
    - ✓ *What are the restrictions on treatment?*
    - ✓ *What contraindications does the medication have?*
    - ✓ *What are the drug interactions?*
    - ✓ *Given certain symptoms and signs, what is the most likely diagnosis?*
    - ✓ *Given a diagnosis and patient information, what treatment should be established?*
    - ✓ *Is it correct to administer metronidazole to a patient who is pregnant since the first trimester of pregnancy?*
    - ✓ *A patient who is pregnant from the first trimester of pregnancy comes with bleeding, is it correct to perform vaginal touch?*
    - ✓ *A 27-year-old male patient who reported itching, foreign body sensation and abundant sparrows in the right eye, which diagnosis is the most likely?*
    - ✓ *A 30-year-old female patient who complains of pain to urination and lower back, which diagnosis is the most likely? What laboratory tests should be requested?*
    - ✓ *Does the result of a laboratory or cabinet study establish or modify the diagnosis?*
    - ✓ *What is the medication and ideal dose for a certain disease?*
    - ✓ *When changing one or several vital signs will the diagnosis or treatment change?*
    - ✓ *When establishing the antecedents, is it associated with the associated diseases?*
    - ✓ *Indicate the type of treatment of first choice in a given disease.*
    - ✓ *Suggest differential diagnoses of a specific disease based on key signs and symptoms.*
    - ✓ *Mention diseases for an age range.*

http://dx.doi.org/10.22201/fca.24488410e.2020.2368

## Ontology Design

In this sub-section we describe the formalization of the main ontology concepts decided for the medical diagnostic requirement. First we introduce the set of terms,

a. Term elicitation consists of producing a seminal list of terms that are relevant for the particular domain of knowledge. The elicitation of terms is done using the list of questions and identifying nouns and verbs. Nouns represent the starting set of concepts that may be used as candidate ontology modules. During term elicitation, there is no need to observe grammatical structure of sentences, only select nouns. The list of terms should be without repetitions and in singular. During the final evaluation of the resulting ontology, this seminal list of terms is used to verify concept coverage (see Table 1).

Table 1
List of terms

| | | | |
|---|---|---|---|
| antecedents | drug interaction | patient | treatment |
| diagnosis | laboratory test | restrictions | vital signs |
| disease | medicament | symptoms | signs |

Source: Author's own

b) Modules identification consists of defining the set of individual ontologies that will conform the ontology system. In order to produce this set of ontology modules, it is necessary to group similar terms by using as input the list of terms, the ontology engineer together with the group of domain expert's workout the clustering of terms. The resulting global scheme of the ontologies that will integrate the ontology system is shown in Figure 2.

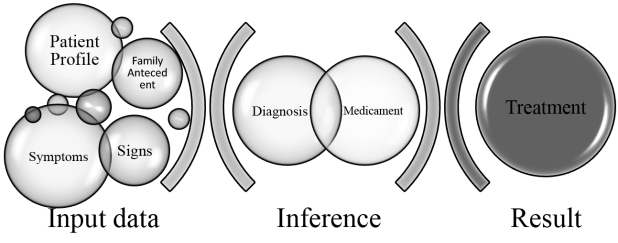


Figure 2. Phases of the diagnosis and treatment of a patient with a disease.
Source: Author's own

http://dx.doi.org/10.22201/fca.24488410e.2020.2368

c) For each of the ontology modules identified, their hierarchies, data properties and object properties are designed and formally stated using Description Logics notation. The following are the main ontology concepts defined:

1) *Medical Antecedent* is an ontology defined to represent the genetic factors, genetic or acquired factors that predisposes a patient to illness. *Antecedents* are defined as something that has a name and description. *Antecedents* are sub classified as *Androgenic, FamilyInheritance, Gynecobstetric, and Personal*. Figure 3 shows the class hierarchy of *Antecedent ontology*.

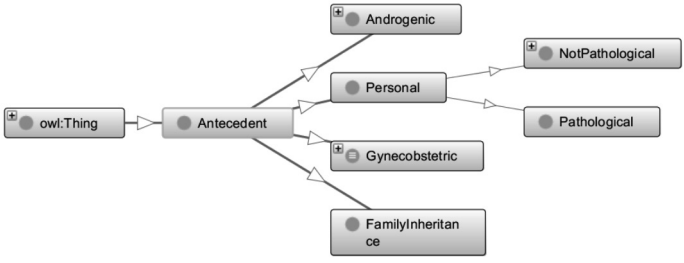


Figure. 3. Subclasses of the Antecedent ontology.
Source: Author's own elaborated with Protégé

The formal definition of the Antecedent ontology is described in DL as follows:

$$TBox = \left\{ \begin{array}{c} Antecedent \equiv \\ \exists hasName.xsd{:}string \sqcap \\ \exists hasDescription.xsd{:}string, \\ Androgenic \subseteq Antecedent, \\ FamilyInheritance \subseteq Antecedent, \\ Gynecobstetric \subseteq Antecedent, \\ Personal \subseteq Antecedent, \\ Pathological \subseteq Personal, \\ NotPathological \subseteq Personal \end{array} \right\}$$

http://dx.doi.org/10.22201/fca.24488410e.2020.2368

2) The purpose of the ontology of *Diagnostic Tests* is the representation of the different laboratory tests that are performed on the patient in order to determine or confirm a diagnosis.

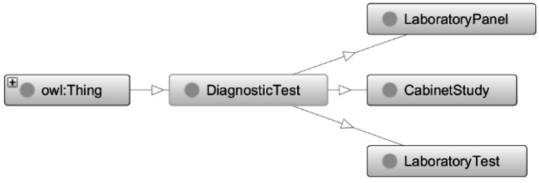


Figure. 4. Subclasses of the Diagnostic Test ontology.
Source: Author's own elaborated with Protégé

The formal definition of the Diagnostic Test ontology is described in DL as follows:

$$TBox = \left\{ \begin{array}{c} DiagnosticTest \equiv \\ \exists hasTestName.xsd{:}string \sqcap \\ \exists hasTestDescription.xsd{:}string, \\ isComposedOf(DiagnosticTest, LaboratoryTest), \\ \forall isComposedOf.LaboratoryTest, \\ LaboratoryPanel \subseteq DiagnosticTest, \\ LaboratyTest \subseteq DiagnosticTest, \\ CabinetStudy \subseteq DiagnosticTest \end{array} \right\}$$

3) The ontology of medical diseases is used to represent the taxonomy of diseases according to the international classification of diseases (CIE 10). Figure 5 shows the class hierarchy of the Disease ontology.

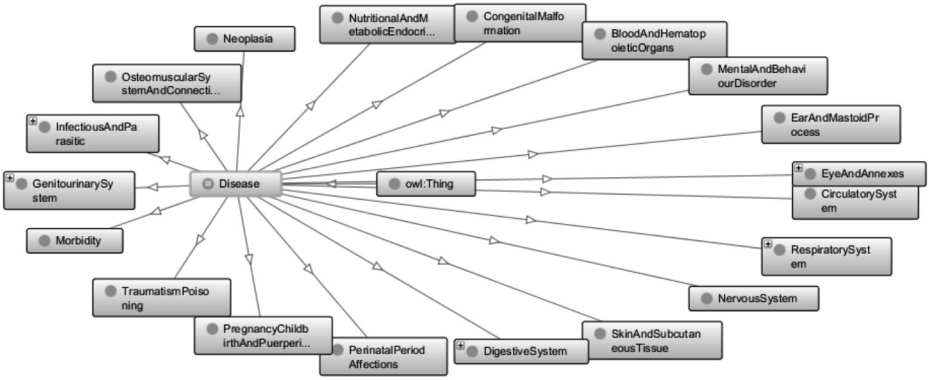


Figure. 5. Subclasses of the Disease ontology.
Source: Author's own elaborated with Protégé

http://dx.doi.org/10.22201/fca.24488410e.2020.2368

$$TBox = \left\{ \begin{array}{c} Disease \equiv \\ \forall hasDiseaseName.xsd{:}string \sqcap \\ \forall hasCIEId.xsd{:}string, \\ CirculatorySystem \subseteq Disease, \\ CongenitalMalformation \subseteq Disease, \\ DigestiveSystem \subseteq Disease, \\ InfectiousAndParasitic \subseteq Disease, \\ Morbidity \subseteq Disease, \\ Neoplasia \subseteq Disease, \\ NervousSystem \subseteq Disease, \\ NutritionalAndMetabolic \subseteq Disease, \\ RespiratorySystem \subseteq Disease, \\ SkinAndSubcutaneousTissue \subseteq Disease \end{array} \right\}$$

4) The *Medicament* ontology is intended to represent the full range of substances and drugs used in the treatment of diseases. Every member of the Medicament class must have an active substance, a presentation, a route of administration and a medical interaction with other medicaments. Figure 6 shows the class hierarchy of the Medicament ontology.

$$TBox = \left\{ \begin{array}{c} Medicament \equiv \\ \forall hasActiveSubstance.xsd{:}string \sqcap \\ \forall hasPresentation.xsd{:}string \sqcap \\ \forall hasRoutOfAdministration.xsd{:}string \sqcap \\ \forall hasMedicalInteraction.Medicament \end{array} \right\}$$

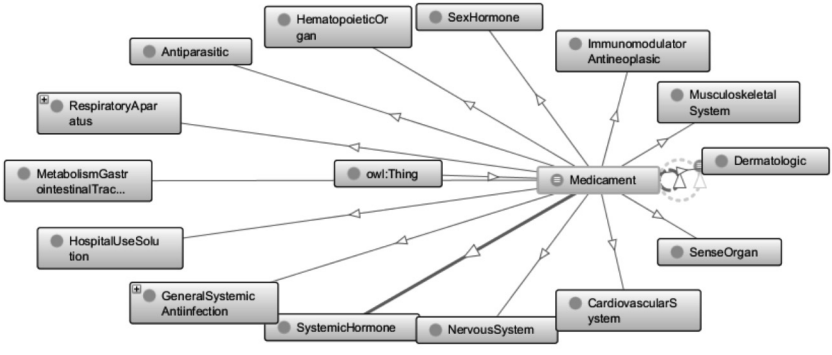


Figure. 6. Subclasses of the Medicament ontology.

Source: Author's own elaborated with Protégé

*5) Patient* ontology aims at representing a person who requires medical care by a physician. In order to represent a patient, the following considerations are met. First every *Patient* must be of two types *Registered* or *Unregistered*. The registered *Patient* is described by a full list of properties.

$$TBox \equiv \begin{Bmatrix} RegisteredPatient \subseteq Patient, \\ UnregisteredPatient \subseteq Patient \end{Bmatrix}$$

$$TBox \equiv \begin{Bmatrix} RegisteredPatient \equiv \\ \forall hasPatientName.xsd:string \sqcap \\ \forall hasLastName.xsd:string \sqcap \\ \forall hasBirthDate.xsd:date \sqcap \\ \forall hasGender \in \{\text{Female,Male}\} \sqcap \\ \forall hasCURP.xsd:string \sqcap \\ \forall hasLifeCondition \in \{Alive, Dead\} \end{Bmatrix}$$

a. Symptoms represent any subjective evidence of disease, while a sign is any objective evidence of disease. Therefore, a symptom is experienced by the patient affected by the disease, while a sign can be detected by different means. In order to represent signs and symptoms the following definitions were made.

$$TBox = \begin{Bmatrix} VitalSign \equiv \\ \begin{Bmatrix} heartRate, breathingFrequency, \\ bloodPressure, temperture, pulse \end{Bmatrix}, \\ VitalSign \subseteq Sign, \\ ClinicSign \subseteq Sign, \\ Somatometry \subseteq Sign \end{Bmatrix}$$

$$TBox = \begin{Bmatrix} Symptom \equiv \\ \forall hasDescription.xsd:string \sqcap \\ \forall hasSymptomName.xsd:string \sqcap \\ Symptom \subseteq SignSymptom \end{Bmatrix}$$

## Ontology Construction

In this sub-section we describe the implementation of the ontology modules and their further integration into the ontology system.

a. Ontology modules implementation. The ontology modules were implemented using the Protégé ontology editor, and represented using the standard Web Ontology Language (OWL). The ontology model consists of a set of imported, independent and

http://dx.doi.org/10.22201/fca.24488410e.2020.2368

auto-contained ontologies among which a set of semantic relations are defined for particular purposes.

b. Ontology modules population. The ontology modules were populated and integrated into the ontology system. The resulting ontology system is composed of 2201 axioms, 195 classes, 22 object properties, 46 data properties, and 310 individuals. The level of DL expressivity is ALCHOIN(D), Figure 7 shows the ontology metrics.

Ontology metrics:

**Metrics**

| | |
|---|---|
| Axiom | **2201** |
| Logical axiom count | **1618** |
| Declaration axioms count | **575** |
| Class count | **195** |
| Object property count | **22** |
| Data property count | **46** |
| Individual count | **310** |
| Annotation Property count | **4** |
| DL expressivity | ALCHOIN(D) |

Figure. 7. Ontology system metrics.

Source: Author's own elaborated with Protégé

c. Ontology system integration. The integrated ontology model consists of the following core ontologies: Patient ontology, Antecedent ontology, Disease ontology, Medicament ontology and Sign and Symptom ontology (see Figure 8).

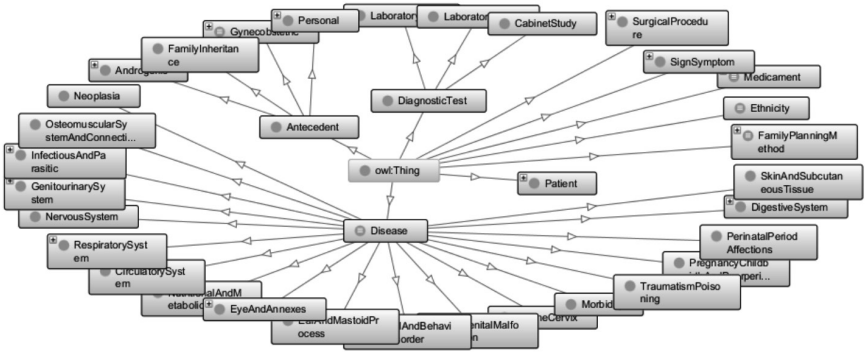


Figure 8. Ontology system integration.

Source: Author's own elaborated with Protégé

http://dx.doi.org/10.22201/fca.24488410e.2020.2368

## Ontology Evaluation

The goal of ontology evaluation is to prove compliance of the world model with the world modelled formally.

a. Evaluation of the *Competency* of the Ontology. The ontological model reported in this work considers the following conceptual dimensions: patient, medicament, disease, sign and symptom, antecedents, and diagnostic tests. All these dimensions correlate and intersect at particular points. In this sub section we describe two scenarios to evaluate the competency of the ontology system reported.

*Scenario 1*. Patient Maria Hernandez is 56 years old, is overweight, has blood pressure of 140/90, her family history includes heart disease and visited the doctor because she has a strong pain in the chest (see Figure 9). Recommended tests: lipid panel, triglyceride, and troponin. Probable diagnostic is: heart disease.

In order to inference the tests recommendations, the following rule is fired:

```
patient:Patient(?p) ^ patient:hasAge(?p, ?age) ^
swrlb:greaterThan(?age, 50) ^ medicaldiagnosis:hasSign(?p,
signsymptom:highBloodPressure) ^
medicaldiagnosis:hasSymptom(?p, signsymptom:overweight) ^
medicaldiagnosis:hasSymptom(?p, signsymptom:anginaPectoris)
->
medicaldiagnosis:hasRecommendedTest(?p,
labtest:LipidProfile) ^
medicaldiagnosis:hasRecommendedTest(?p,
labtest:TriglyceridesTest) ^
medicaldiagnosis:hasRecommendedTest(?p,
labtest:TroponinTest)
```


Figure 9. Data properties and object properties of patient Maria Hernandez.

Source: Author's own elaborated with Protégé

http://dx.doi.org/10.22201/fca.24488410e.2020.2368

Another form to generate laboratory tests recommendations is by defining a *PatientTestRecommendation* class sub-classified into *LipidProfileTest*, *GlucoseTest*, etc. For example, the *LipidProfileTest* is formally defined as follows:

$$TBox = \left\{ \begin{array}{c} LipidProfileTest \subseteq PatientTestRecommendation \sqcap \\ LipidProfileTest \equiv Patient \sqcap \forall hasSign.highBloodPressure \sqcap \\ \forall hasSymptom.overweight \sqcap \forall hasSymptom.anginaPectoris \end{array} \right\}$$

As a result of this axiom definition, the reasoned adds Patient Maria Hernandez to the *LipidProfileTest* class (see Figure 10).

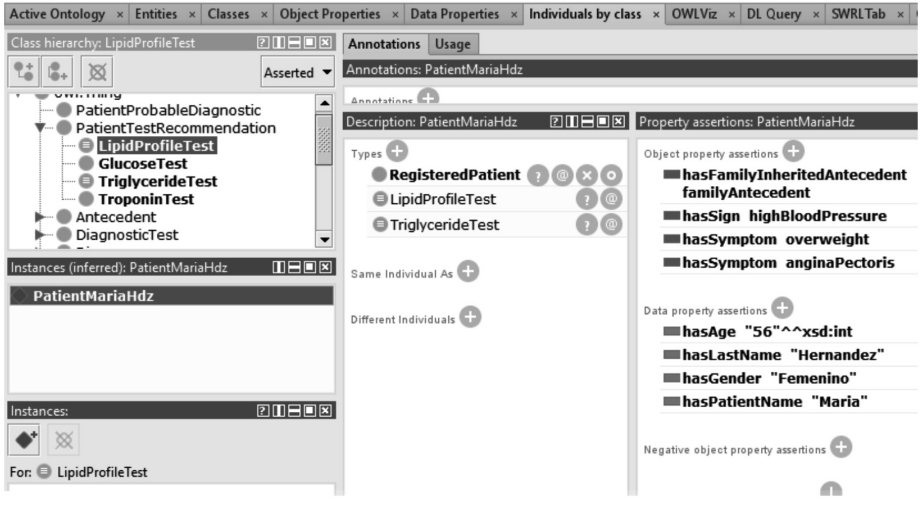


Figure 10. Recommended tests inferred for the registered patient Maria Hernandez based on her signs and symptoms.

Source: Author's own elaborated with Protégé

*Scenario 2*. Ramon Perez is an 11 years old child that has breath difficulty, cough, mucus production, blueness of the lips, lack of energy and wheezing. Recommended tests: Blood gas test, bacterial culture, and viral culture. Diagnostic: chronic bronchitis (see Figure 11).

http://dx.doi.org/10.22201/fca.24488410e.2020.2368

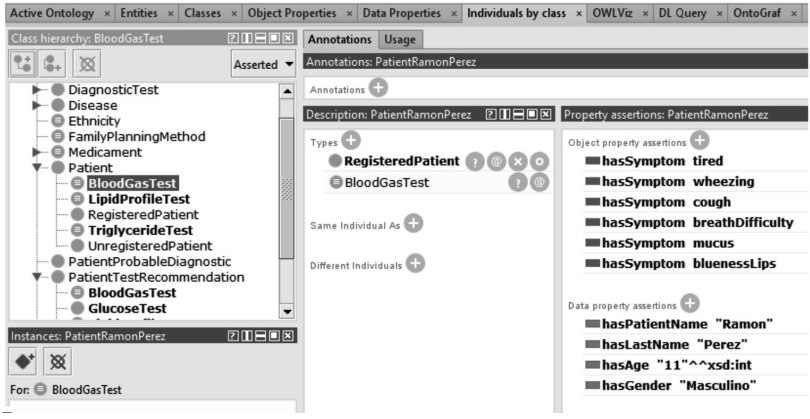


Figure 11. Recommended tests inferred for the registered patient Ramon Perez based on his signs and symptoms.

Source: Author's own elaborated with Protégé

b. Evaluation of the Quality. To evaluate the quality of the ontology system, the following design principles were considered: clarity, coherence and modularity.

1. *Clarity*. In order to address this design principle, the ontology concepts were implemented using formal axioms, and when possible complete definitions were created, complete definitions are represented as formal equivalences. However, there were various cases where the incorporation of complete definitions made the ontology to expand in memory demand. Figure 12 shows the main concepts defined for the integrated ontology system. Complete definitions were defined for: *Disease*, *Ethnicity*, *Family Planning Method*, and *Medicament*.

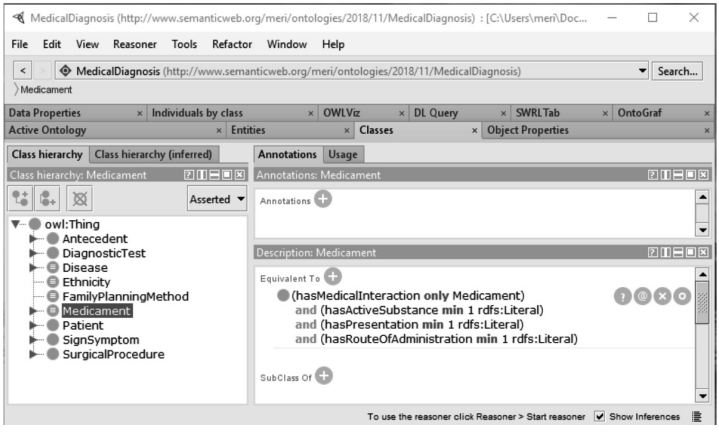


Figure 12. Complete definitions incorporated in the ontology system.

Source: Author's own elaborated with Protégé

http://dx.doi.org/10.22201/fca.24488410e.2020.2368

2. *Coherence*. Coherence was verified by executing the reasoning tasks of consistency checking and classification. Figure 13 shows the result of executing Pellet reasoning services of class hierarchy, object property hierarchy, data property hierarchy, class assertions, object property assertions, and same individuals returned a consistent ontology model.

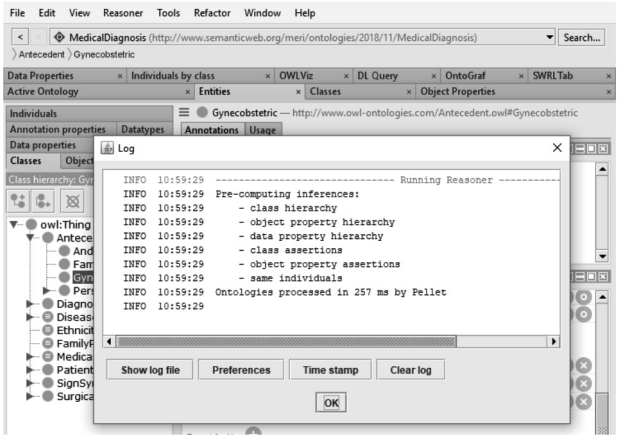


Figure 13. Execution of Pellet reasoning tasks.

Source: Author's own elaborated with Protégé

3. *Modularization* of the ontology model reported in this paper was achieved by dividing the model into six independent ontologies. Each of these ontologies is domain independent and is reusable. For instance, the *Patient* ontology can be reutilized and maintained by the social services department and several applications could be useful, such as patient remote assistance, a system to control and supervise drug administration to patients.

The most important design principles were considered and verified through Protégé tools such as reasoners and DLQuery. Ontology consistency checking was executed to verify that none of the class definitions and axioms had logical contradictions, or the individual's instantiated into the ontology.

## Analysis of Related Methodologies

An analysis of the most cited and popular ontology design methodologies is described in this section. The analysis of ontology building methodologies focuses specifically on reviewing whether methodologies consider the following characteristics: modular-design,

domain-oriented design, user-centered design, the type of development process (iterative and/or incremental), the incorporation of design principles, and competency-based evaluation.

Lenat and Guha (1989) described the methodology they followed to build the Cyc ontology at the Microelectronics and Computer Technology Corporation. The Cyc ontology represents an important attempt to codify a large amount of common-sense knowledge. Authors describe a method composed of three processes to build the ontology: manual extraction and codification of knowledge, computer-human aided extraction and codification of knowledge and full computer aided codification of knowledge. The methodology of the Cyc project is not oriented to answer a determined collection of competency questions, it rather covers a wide spectrum of common-sense knowledge.

In 1995, Grüninger and Fox (1994) and (1995) described the process of building an ontology for the Toronto Virtual Enterprise (TOVE) modelling project. The TOVE ontology was constructed with the objective of representing a common-sense enterprise model. The methodology defined the following procedure: Describe a motivating scenario, define the competency questions, define the terminology of the ontology (objects, attributes and relations), and specify the definitions and constraints on the terminology specification, and evaluation of the ontology by completeness theorems. The specific ontology design principle that authors address with this methodology is ontology completeness based on competency questions. Even though that authors describe a two ontology example (composed of Activity and Organization ontologies), they do not provide specific guidelines for the application of ontology for design principles.

Uschold and King (1995) presented a methodology for developing and evaluating ontologies, later refined by Uschold and Grüninger (1996). This methodology included the following stages to build ontologies: identify the purpose, build the ontology - ontology capture, ontology coding and integration of existing ontologies-, evaluation, and documentation. They incorporated a scoping process during the ontology capture stage which includes grouping. Grouping was defined as the task of structuring terms into work areas corresponding to naturally arising sub-groups. In this sense, grouping is the closest concept to the ontology modularization. However, the Uschold and colleague's methodology does not cover integration guidelines for networked ontologies or ontology systems. Ontology evaluation is not clearly defined nor practical guides or techniques are provided.

Bernaras et.al (1996) presented a methodology for building ontologies under the project kactus. The main objective of authors when proposing this methodology was to evaluate the feasibility of knowledge reuse in complex systems. Their methodology stated the following general processes: provide a specification of the application to know the application context and a view of the components; produce a list of terms and tasks; make a preliminary design based on the relevant top-level ontological categories taking as input the

list of terms and tasks developed during the previous phase; augment the domain concepts and relations identified during the previous step; refine and structure the ontology in order to make a definitive design. The KACTUS methodology main objective is very similar to the approach presented in this paper, which is to promote ontology reuse throughout the ontology design process. This methodology does not describe a clear evaluation process.

The methontology framework was presented for the first time by Gómez-Pérez (1996). It was later refined and detailed by Fernández et.al (1997) and (1999), Gómez-Pérez (1998), to facilitate the construction of ontologies. methontology defines a development process and life cycle which consists of the following activities: specification, conceptualization, formalization, implementation, and maintenance. Of special interest is the conceptualization activity which sets the following tasks: to build the glossary of terms, to build concept taxonomies, to build binary relation diagrams, to build the concept dictionary, to describe in detail each binary relation, to describe in detail each instance attribute, and to describe in detail each constant. The conceptualization phase of this methodology involves the documentation of intermediate representations that turns out to be an excessive and tedious work for very large ontologies.

CommonKADS (Schreiber, 2000) is an important methodology for knowledge-based systems construction, not specifically tailored for the design and construction of ontologies. However, CommonKADS is a methodology which includes the concept of model components for reuse. This methodology recognizes the need for reusing knowledge-model elements or a combination of them, considering that large parts of a given model are not specific and may re- occur in other domains and/or tasks. CommonKADS methodology includes the following steps for knowledge-based systems: knowledge identification, consisting of getting familiarized with information sources, glossary and scenarios, and identifying model components for reuse; knowledge specification, aims at constructing a knowledge specification of the knowledge model by choosing the task template and the initial domain conceptualization; and knowledge refinement, which validates the knowledge model and completes the knowledge bases.

Noy and McGuinness presented in (2001) a methodology to build ontologies consisting of the following steps: determine the domain and scope of the ontology, reuse existing ontologies, enumerate important terms, define classes and class hierarchy, define properties of classes, define facets of properties, and create instances. Even though this methodology is well explained, it does not include an evaluation step.

The NEON methodology (Suárez-Figueroa, Gómez-Pérez, & Fernández-López, 2012) is based on the use of ontology design patterns (ODP). This methodology establishes the reutilization of ontologies from a public ontology repository and a set of known ontology design patterns to integrate them by means of a re-engineering process. The general steps

defined in this methodology are: 1) identify requirements, 2) identify available design patterns, 3) divide and transform the selected problem into partial problems, 4) match selected partial problems with ontology design patterns, 5) select the design pattern, 6) apply selected patterns to make a composition, 7) evaluate partial designs solutions, and 8) integrate partial solutions. The NEON methodology depends on the existence of a repository of ontology common problems and a collection of ontology design patterns associated with general use cases. When the users of the ontology specify the set of competency questions at the beginning of the methodology, these questions need to be associated with the general use cases. Authors propose the utilization of additional tools to support the end users in the validation and correlation of competency questions.

According with Gangemi and Presutti (2009) a design pattern provides a modelling solution to a frequent ontology design problem. Authors define Content Ontology Design Patterns as small ontologies that mediate between problem types and design solutions and are used as modelling components to the extent that a new ontology can be built from the composition of multiple components.

Table 2
Comparison of related methodologies

| Methodology | User-centered design | Modular oriented | Domain oriented | Incremental and iterative | Quality oriented | Competency-based evaluation |
|---|---|---|---|---|---|---|
| Lenat and Guha (1989) | No | Yes | No | Yes | No | No |
| Grüninger and Fox (1995) | Yes | No | Yes | No | Yes | Yes |
| Uschold and King (1995) Uschold and Grüninger (1996) | Yes | Yes | Yes | No | Yes | Yes |
| Bernaras et.al (1996) | No | Yes | Yes | No | Yes | No |
| METHONTOLOGY | Yes | Yes | Yes | Yes | Yes | Yes |
| CommonKADS | No | Yes | Yes | Yes | No | No |
| Noy and McGuinness (2001) | Yes | No | Yes | Yes | No | No |
| NEON | Yes | Yes | Yes | Yes | No | Yes |

As can be seen in Table 2, the most complete methodologies are METHONTOLOGY and NEON. However, METHONTOLOGY generates an overload of effort for the documentation

during design and development of very large ontology systems. The NEON methodology depends on the existence of a repository of ODPs associated with general use cases, end users of the NEON methodology will require extra effort to identify the general use cases that address their particular requirements.

## Conclusions

A main problem in Decision Sciences is the formal representation of complex problems with large volume of data, the methodology presented in this paper provides ontologies design principles that could be applied to construct ontological models and formulate scenarios to achieve improvements in decision making process.

This paper promotes the reutilization of ontologies by implementing ontology modules from the beginning of the ontology design, ontologies are seen as reusable modules not as general design patterns. The idea behind this approach is that the owners of resulting ontology modules can reuse them inside the enterprise for more applications, being the designers of the structures of ontologies they will be able to reuse them easily instead of searching in a given repository for general design patterns.

The incorporation of ontology design patterns is a good approach whenever the set of required solutions exist in a public available repository. However, many times ontology design requires the implementation of tailored constructs for complex engineering systems and heterogeneous domains. Even more, ontology reuse requires the adaptation of the pre-existing ontology design solution to the specific application needs. The notion of ontology design pattern is close to the concept of modularized ontology design, in the sense of reusability of ontology modules and composition of new ontologies based on a set of initial components.

## References

Bernaras, A., Laresgoiti, I., & Corera, J. (1996). Building and Reusing Ontologies for Electrical Network Applications. In ECAI (pp. 298-302). PITMAN.

Fernández, M.; Gómez-Pérez, A.; Juristo, N. (1997) METHONTOLOGY: From Ontological Art Towards Ontological Engineering. Symposium on Ontological Engineering of AAAI. Stanford (California). Disponible en: http://oa.upm.es/5484/1/METHONTOLOGY_.pdf. Consultado: 24/08/2019.

Fernández, M.; Gómez-Pérez, A.; Pazos-Sierra, J. (1999) Building a chemical ontology using METHONTOLOGY and the ontology design environment, *IEEE Intelligent Systems & their applications 4* (1) 37–46. https://doi.org/ 10.1109/5254.747904

Gangemi, A., & Presutti, V. (2009). Ontology design patterns. In *Handbook on ontologies* (pp. 221-243). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-92673-3_10

Gómez-Pérez, A. (1994). From knowledge based systems to knowledge sharing technology: Evaluation and assessment. Disponible en: http://oa.upm.es/6498/. Consultado: 24/08/2019.

Gómez-Pérez, A. (1999). Ontological engineering: A state of the art. Expert Update: *Knowledge Based Sys-*

*tems and Applied Artificial Intelligence, 2*(3), 33-43.

Gómez Pérez, A. Knowledge Sharing and Reuse (1998). In J. Liebowitz (Editor) *Handbook of Expert Systems*. CRC.

Gómez-Pérez, A., Fernández, M., & Vicente, A. D. (1996). Towards a method to conceptualize domain ontologies. 12th European Conference on Artificial Intelligence. http://oa.upm.es/7228/

Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing. Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University.

Gruninger, M., & Fox, M. S. (1994). The design and evaluation of ontologies for enterprise engineering. In Workshop on Implemented Ontologies, European Conference on Artificial Intelligence (ECAI).

Grüninger, M., & Fox, M. S. (1995). Methodology for the Design and Evaluation of Ontologies. Proc. of the IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing.

Lassila, O., & McGuinness, D. (2001). The role of frame-based representation on the semantic web. *Linköping Electronic Articles in Computer and Information Science, 6*(5), 2001.

Lenat, D. B., & Guha, R. V. (1989). *Building large knowledge-based systems; representation and inference in the Cyc project*. Addison-Wesley Longman Publishing Co., http://portal.acm.org/citation.cfm?id=575523

Neches, R., Fikes, R. E., Finin, T., Gruber, T., Patil, R., Senator, T., & Swartout, W. R. (1991). Enabling technology for knowledge sharing. *AI magazine, 12*(3), 36. https://doi.org/10.1609/aimag.v12i3.902

Rector A. L. Modularisation of Domain Ontologies Implemented in Description Logics and related formalisms including OWL. *Proceedings of the 2nd International Conference on Knowledge Capture*. 2003: 121-128. https://doi.org/10.1145/945645.945664

Schreiber, G. (2000). *Knowledge engineering and management: The CommonKADS methodology*. MIT press. https://doi.org/10.1016/s0898-1221(00)90199-8

Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology.

Suárez-Figueroa, M. C., Gómez-Pérez, A., & Fernández-López, M. (2012). The NeOn methodology for ontology engineering. In Ontology engineering in a networked world (pp. 9-34). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-24794-1_2

Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. *Knowledge engineering review, 11*(2), 93-136. https://doi.org/10.1017/S0269888900007797

Uschold, M., & King, M. (1995). Towards a methodology for building ontologies (pp. 15- 30). *Edinburgh: Artificial Intelligence Applications Institute*, University of Edinburgh.