REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : IA13 /M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Intelligence Artificielle (IA)**

Développement d'un système d'aide au diagnostic médical en utilisant les ontologies floues

Par :

CHAIMA ZIAD

Soutenu le /07/2021, devant le jury composé de :

Nom et prénom	Grade	Président
Ben Seghier Nadia	MCA	Rapporteur
Nom et prénom	Grade	Examinateur

Année universitaire 2020-2021

ACKNOWLEDGEMENTS

I wish to thank my committee members who were more than generous with their expertise and precious time.

A special thanks to Mme *"Nadia Ben Seghier"* my supervisor for her countless hours of reflecting, reading, encouraging, and most of all patience throughout the entire process.

many thanks for everyone and a lot of respect .

DEDICATION

I dedicate my work to my family,

A special feeling of gratitude to my loving parents, "*Aicha Maafi*" and "*Mostafa Ziad* " for thier Greate efforts to Make me who I'm today , for the encouragement ,for their blessings and prayers which were in my company the whole life ,for many uncountable things.

I dedicate this work for my beloved Grandmother, may Allah rest her soul.

For my dear little sisters *Asma, Rayane* and *Hadjer* as well.

I also dedicate this work to my many friends and classmates who have supported me in my difficult moments and stands with me all the time.

"Rania Cherif, Saadi Sabrina, Ghamrani Chaima, Zineb Hdaga, Imane Lguess, Imane Chabi, Hana Bouaziz, Halima Khineche, Rokia Rahmani, Gidouam Afaf, Nesrine Amrat, Zouzal Khadija, Douaa Benchiha, Salsabil Zemmal, Insaf Slimani".

To my sweet neighbor "*Jasmine Madhoui*" and my dear cousint "*Safa Mouaki dadi*".

I dedicate it as well for some other dear friends that I still love but I can't be a friend with anymore.

I will always appreciate all they have done.

I dedicate this work and give special thanks to my dear best friend

"*Rahma Soltani* " who always gave me precious invaluable things and life lessons as a gifts, wishing her all the best in her life .

All the love for them and respect.

Special thanks for my competitor's classmates and teachers who have done their best to obstruct my studies in unethical ways but they prove me at the end that it is impossible and I'm strong enough to deal with all of them on my own.

Abstract

In order to build a robust medical diagnosis system witch is able to know the right disease between many diseases with similar symptoms, in this work we have created a fuzzy ontology based on fuzzy logic to deal with the inaccuracies and uncertainties that characterize certain concepts and relationships, where the role behind it is to assist our system. We have created a java program to manage this ontology as a knowledge base for this program and to use it in easy way as well.

Keywords: Fuzzy ontology, Fuzzy logic, Medical diagnosis.

Résumé

Afin de construire un système de diagnostic médical robuste capable de connaître la bonne maladie entre de nombreuses maladies présentant des symptômes similaires, dans ce travail, nous avons créé une ontologie floue basée sur la logique floue pour traiter les inexactitudes et les incertitudes qui caractérisent certains concepts et relations, où le rôle derrière cela est d'assister notre système. Nous avons créé un programme Java pour gérer cette ontologie en tant que base de connaissances pour ce programme et pour l'utiliser de manière simple également.

Mots clés : Ontologie floue, Logique floue, Analyse médicale.

Contents

List of Figures

List of Tables

General Introduction

One of the challenging directions of the research is the knowledge representation in medicine field in general including medical diagnosis field which is our topic for this project.

By the computer science advancement this field has been advanced a lot. Lately the Ontology has been largely used especially in the medical diagnosis field because of a plenty of advantages suited such as: knowledge bases updating, data warehouse organizing, the integration of data between systems, making decision and access to relevant web information.

The type of information treated in medical diagnosis is most often imperfect : ambiguity, inaccuracy and uncertainty, these are the imperfections we are talking about and that's exactly the type of information treated by the medical diagnosis field, so by the result this is the kind of knowledge we need to represent, and that's absolutely never been a simple task.

In fact, there are several methods and techniques have been proposed to solve this problem, the main is the Fuzzy logic.

Fuzzy Logic is a method of reasoning that resembles human reasoning. Its approach imitates the way of decision making in humans that involves all intermediate possibilities between digital values YES and NO. What fuzzy logic offers to our system is the ability to express knowledge in a linguistic way, which makes it possible to describe a system using a simple and friendly rule. Finally, as a result of the natural integration of fuzzy logic in ontology a new theoretical paradigm called fuzzy ontology has been defined, and that's actually what we are going to build in our project.

This report presents the design and the implementation of fuzzy ontology based medical diagnostic help system, in order to build a robust medical diagnosis system witch is able to know the right disease between many diseases with similar symptoms.

Our fuzzy ontology represents and manipulates the knowledge and data relating to the diagnosis of pneumonia disease, influenza and Covid-19 while taking into consideration the fuzzy aspects (imprecision and uncertainties).

In this work, we have also created a java program to manage this ontology as a knowledge base for this program and to use it in easy way as well.

Chapter I

The Ontologies

1 Introduction

Humans require words (or at least symbols) to communicate efficiently. The mapping of words to things is indirect. We do it by creating concepts that refer to things, by using ontologies we're able to represent knowledge with its semantics in computer science either. Ontologies are actually a source of knowledge and it is the infrastructure for the semantic web as well, that's why they are most used in too many domains such as knowledge representation. Also it has its roots as a field of philosophical study that is focused on the nature of existence.

However, today's ontology can incorporate computable descriptions that can bring insight in a wide set of compelling applications including more precise knowledge capture, semantic data integration, sophisticated query answering, and powerful association mining there by delivering key value for health care and the life sciences. This chapter introduces ontologies: what they are, their usefulness, their components besides to their types and examples and how they can be developed.

2 The definition of ontologies

2.1 From philosophy

Ontology is the study of being. It focuses on several related questions:

- What things exist?
- What categories do they belong to?
- Is there such a thing as objective reality?
- What does the verb “to be” mean?

Some of these questions may seem painfully abstract and not very useful, but they are and always have been enormously important to some philosophers, especially to those who believe in foundationalism.

Foundationalist philosophers believe that to arrive at truth it is necessary to start with the most fundamental issues to be sure about the foundations of philosophy and then work our way up from there to more specific questions. If you believe in foundationalism, then probably the most important questions are ontological questions. [1]

2.2 History

Since the mid-1970s, researchers in the field of artificial intelligence (AI) have recognized that knowledge engineering is the key to building large and powerful AI systems.

AI researchers argued that they could create new ontologies as computational models that enable certain kinds of automated reasoning, which was only marginally successful. 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-based systems.

In particular, David Powers introduced the word ontology to AI to refer to real world or robotic grounding, publishing in 1990 literature reviews emphasizing grounded ontology in association with the call for papers for a AAAI Summer Symposium Machine Learning of Natural Language and Ontology, with an expanded version published in SIGART Bulletin and included as a preface to the proceedings. Some researchers, drawing inspiration from philosophical ontologies, viewed computational ontology as a kind of applied philosophy.

In 1993, the widely cited web page and paper "Toward Principles for the Design of Ontologies Used for Knowledge Sharing" by Tom Gruber used ontology as a technical term in computer science closely related to earlier idea of semantic networks and taxonomies. [2]

2.3 Definition

Definition 1: [3]

- Etymological origin: from Greek (ontos = being, logia = speech).
  Semiotic point of view on ontology: Semiotics: studies the process of meaning, that is to say the production, codification and communication of signs.
  Syntax: a set of signs (symbols, lexical entries) which make sense for humans, not machines.
  Semantics: relationships between signs and real world entities.
  Pragmatics: what signs are used for what purpose? Social: who uses which signs?
- "In AI, an ontology represents an engineering artifact, made up of a specific vocabulary used to describe a certain reality, accompanied by a set of implicit assumptions regarding the meaning of words in this vocabulary "
- What ontologies are not: a world catalog, a list of objects, only a knowledge base
  What ontologies are: terminologies, systems (structure) of meaning: eg taxonomies
  the purpose of ontologies is to be: consensual, normative, consistent, shareable, reusable.

Definition 2: [4]

An ontology describes the common words, concepts and relationships between concepts used to describe and represent an area of knowledge. An ontology can range from a: Taxonomy (knowledge with minimal hierarchy or a parent/child structure) to a Thesaurus (words and synonyms) to a Conceptual Model (with more complex knowledge) to a Logical Theory (with very rich, complex, consistent and meaningful knowledge).

A well-formed ontology is one that is expressed in a well-defined syntax that has a well-defined machine interpretation consistent with the above ontology definition.

Definition 3: [5]

In 1998, Studer et al. merged these two definitions of Gruber(In 1993) and Borst (In 1997) stating that:“An ontology is a formal, explicit specification of a shared conceptualization.”

3 The usefulness of ontologies

- Data and information system integration
    - Integrating legacy systems
    - Trying to prevent interoperability problems
    - Data-level data integration
- Ontologies as part of a solution to other problems
    - E-Learning
    - Deep question answering with Watson
    - Digital humanities
    - Semantic scientific workflows

4 Components of Ontology

The ontology can be seen as a 5-tuple where its components are: Concepts, relationships, functions, individuals or instances and axioms [6], where:

- **Concepts (classes):** are the main formalized elements of the domain. Since the logic, the concepts can be described using specific properties which must be satisfied by them.
- **Relationships:** are links between the concepts for representing the ontology structure (taxonomic or not taxonomic).
- **Functions:** are elements with the purpose of calculating information from the other elements.
- **Instances (objects):** are the representations about the main objects within domain according to ontology structure.
- **Axioms:** are the restrictions, rules, logic correspondences definitions which must be accomplished in the relationship between the ontology elements. The axioms can be seen as the smallest unit of knowledge within an ontology.

5 Ontology Evaluation [6]

The ontology is a very important tool for the information representation that has become a standard for the knowledge representation, so it is necessary to evaluate its main aspects in order to guarantee that representation to be the most real according to the domain. The evaluation of an ontology can be seen from two approaches: verification and validation, or evaluation by criteria.

5.1 Verification

The verification is used to determine if the ontology is correctly constructed or not by the satisfaction of the competency questions which was defined as part of the proposed task, if the built ontology is suitable for the real world. The verification has the next focus:

- **Lexical**: the lexical verification is related to the vocabulary used for conceptualizing the domain and is regularly made with precision, recall and F-measure. This verification includes aspects related to the reusability.

- **Taxonomic**: specifically, this verification is only focused on is-a and has-a relationships within the ontology.
- **Semantic**: is based on consistency by the semantic features of the ontology considering the meaning and content of these features.
- **Context**: can be evaluated by other web ontologies or specific applications which are in the same domain.
- **Syntactic**: this verification is about the coherence in the ontology definitions. For a deep verification, it is necessary using some criteria.
- **Structural, architectural and design:** have the purpose to identify the absence of the main concepts of a domain in the structure, loops, concepts with the same definition but different name, among other errors. In this kind of verification does not exist metrics well defined because could be ambiguous and normally the metrics are context-free.

5.2 Validation

The validation is about the ontology definitions which should most real possible model and represents a defined domain, indicates if the ontology definitions are a model of the real world. The validation is made by the next techniques:

- Gold standard: In this validation, the ontology is compared with a gold ontology that was created by an expert, where both represent the same domain; it assumes the gold ontology is completed and corrected else the validation could have many errors and is commonly used for ontology which was created by semi-automatic and automatic process
- Application based: the application based validation is a focus for deter- mining if the results of the proposed ontology satisfy or not the task for which it was done.
- Data driven: for applying this validation is necessary to compare the in- formation of the ontology with existing data about the same domain meanly using a corpus of text documents. A very important aspect in this validation is determining if the ontology has enough elements for representing the domain.
- User based: basically, this validation is about the experience and perspective around the final user because it becomes suggestive and empiric; for minimizing the error it is advisable the user be an expert in the ontology domain.

5.3 Evaluation by Criteria

The evaluation by criteria is a focused technique in aspects or features which can be quantifiable in order to ease the requirements analysis and some of them can be measured by ontology tool as reasoners. In the Table 1.1 some criteria are described.

Criteria	**Description**
Lawfulness	About the syntactical error frequency
Richness	Quantifies the most important syntactic features are used in the ontology
Adaptability	How the ontology responses in future uses
Clarity	To communicate about the meaning of the terms independently of the context
Closeness Index	Measures the closeness or structural resemblance among the concepts
Similarity Index	About the proposition correctness into knowledge structure
Coherence	Measure if exist contradictions among the elements of ontology according to logical consistency
Computational Efficiency	Measures the reasoner performance when processing the ontology
Conciseness	About the existence of irrelevant information unnecessary and redundancies
Modularity	Indicates if the ontology was created with an appropriate methodology in order to define if exist reusable components
Tangledness	Measures the distribution of multiple concepts and if exists intersections into the structure
Connectivity	Indicates the most important concepts based on the amount of relationships.
Consistency	Indicates if exists a contradiction among the definitions of the ontological elements
Completeness	About if the content of the ontology is explicit or can be inferred in order to consider it complete
Sensitiveness	Measures how a small change in a definition modifies other elements
Standard Coverage	Identifies elements which are not defined in the ontology
Coupling	About the number of external concepts that are referenced or imported
Coverage	Indicates how well the ontology represents or models the domain

Table 1.1: Criteria for the ontology evaluation [6]

6 Types and Examples of Ontologies [3]

6 .1 Types of ontologies

- **Top-Ontologies**: category ontologies (philosophical)

  - Very high level concept category specification (object, event, state, process, …)
- **Core-Ontologies**: generic ontologies of thesaurus type:

  - Specification of a reference vocabulary (physics, math, …)

  - Hierarchies of terms, semantic relationships between terms (synonyms, composition, ...)
- **Domain ontology**: descriptive ontologies

  - Semantically rich specification of a domain

  - Definitions of concepts and relations between these concepts to build specialized ontologies from more general ontologies (core- ontologies) describing a domain
- **Taxonomies**

  - From the Greek "taxis": arrangement, order (taxon: class)

- Science of the laws of classification

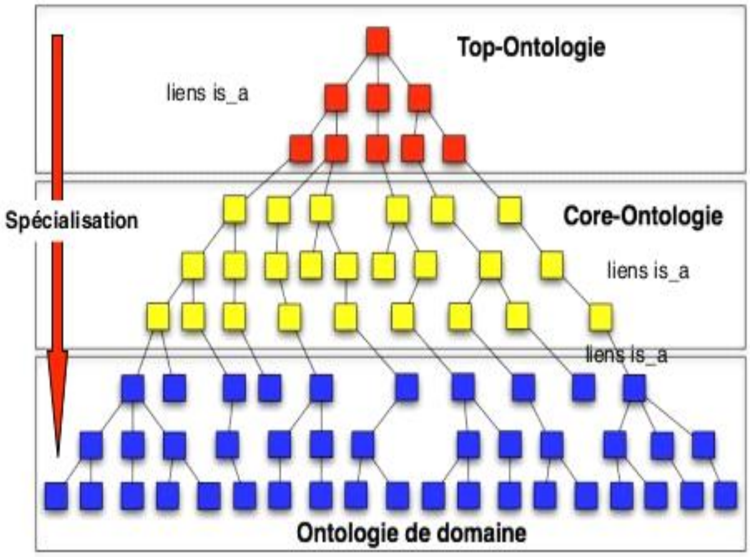


Figure 1.1: Types d'ontologies [3]

6 .2 Examples of ontologies

- **Top-Ontologies:** SUMO(Standard Upper Ontology), DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering),KR Ontology ,Upper Cyc, ... etc
- **Core-Ontologies:** Mikrokosmos , TOVE (University of Toronto), ... etc
- **Domain ontologies:** ontologies diverses (KSL serveur), ...etc
- **Taxonomies:**
  - Taxonomies in life sciences: ex: fauna, flora, ...
  - Taxonomies of the artificial: ex: taxonomy of machining, ... etc

7 Ontological Engineering

7.1 Identification of Objectives and Context

- an ontology for what?

- intended uses, purposes?

- Potential users of the ontology?

- domains, "semantic" coverage: vocation encyclopedic or specialized,

- objectives: inform, standardize, capitalize, research, inter-software communication,

- users: terminologies, sources of information,

- target properties: consensus, coherent, consistent, inferences.

- process: acquisition method, language of representation.

7.2 Construction of Ontology

7.2.1 Capture of Ontology: Independent of a language of "Conceptualization"

- Identification of key concepts and relationships
- Categorization and Capture of Ontology: How to determine the concepts?
    - Top-down approach: start from a reduced number of concepts that we specialize.
    - Bottom-up approach: start from all the specific terms.
- Intermediate Approach (Middle-Out, Basic Level): the concepts are structured around intermediate concepts, neither too general nor too specific.
- Precise & unambiguous LN definitions for concepts and relationships.
- Identification of terms denoting concepts and relationships.
- Approval

7.2.2 Coding of Ontology: "Explicit representation of the Conceptualization"

- Formal language: Prolog, KL-One, OIL, CG.

7.2.3 Integration of Existing Ontologies

- During the Capture and / or Coding phases.
- Difficult problem.

7.3 Assessment

- Confrontation of ontology: objectives, software, users.

7.4 Documentation

- Essential for the acceptance of the ontology

8 Conclusion

Ontologies have been and still a strong factor for AI advancement, by giving sense to every single information in software exactly the same like the human mind.

In this chapter we have introduced the meaning of Ontology from different approaches, their usefulness, their components also, and how can they be evaluated. We have introduced also their types and some examples for them and finally the ontological engineering. In the next chapter, we’re going to introduce the fuzzy logic.

Chapter II

Fuzzy Logic

1 Introduction

Fuzzy logic is an extension of Boolean logic by Lotfi Zadeh in 1965 based on the mathematical theory of fuzzy sets, which is a generalization of the classical set theory. By introducing the notion of degree in the verification of a condition, thus enabling a condition to be in a state other than true or false, fuzzy logic provides a very valuable flexibility for reasoning, which makes it possible to take into account inaccuracies and uncertainties. One advantage of fuzzy logic in order to formalize human reasoning is that the rules are set in natural language.

This chapter introduces Fuzzy logic: the definition, the architecture, its advantages and disadvantages, implementation, their applications.

2 Fuzzy logic definition

Definition 1: [7]

In fuzzy mathematics, fuzzy logic is a form of many-valued logic in which the truth values of variables may be any real number between 0 and 1 both inclusive. It is employed to handle the concept of partial truth, where the truth value may range between completely true and completely false. By contrast, in Boolean logic, the truth values of variables may only be the integer values 0 or 1.

Fuzzy logic is based on the observation that people make decisions based on imprecise and non-numerical information. Fuzzy models or sets are mathematical means of representing vagueness and imprecise information (hence the term fuzzy). These models have the capability of recognizing, representing, manipulating, interpreting, and utilizing data and information that are vague and lack certainty.

Definition 2: [8]

“Fuzzy logic” is often understood in a very wide sense which includes all kinds of formalisms and techniques referring to the systematic handling of degrees of some kind. In particular in engineering contexts (fuzzy control, fuzzy classification, soft computing) it is aimed at efficient computational methods tolerant to sub-optimality and imprecision.

Definition 3: [9]

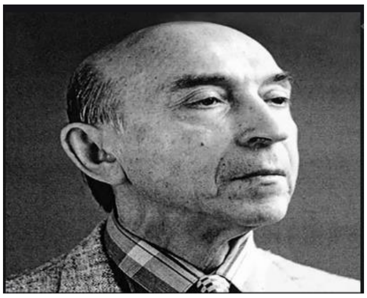


Figure 2.1: Lotfi Zadeh

The term Fuzzy Logic was first described by **Lotfi Zadeh** in 1965. He thought that traditional computer logic is not capable of handling unclear or vague data. Similar to humans, there are many possible values between True and False that a computer can incorporate. **These can be: Certainly yes ,Possibly yes ,Can't say ,Possibly no ,Certainly no .**

For example: Problem – Is it hot outside?

Solution:

Boolean Logic	**Fuzzy Logic**
Yes (1.0)	Very hot (0.9)
No (0)	Little hot (0.20)
	Moderately hot (0.35)
	Not hot (1.0)

3 Fuzzy sets [10]

- The classical set theory simply designates the branch of mathematics that studies sets. For example, 5, 10, 7, 6, 9 is a set of integers. 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 is the set of integers between 0 and 10. 's' 'd'; 'z', 'a' is a set of characters. "Site", "of", "zero" is a set of words. Usually, a set is denoted by a capital letter: thus, we write A = 6,7, 9. The empty set is denoted: it is a remarkable since it contains no element. This seems unnecessary at first glance, but in fact, we will often use it. Sets are often represented in graphic form, typically by circles, as figure 2.2 illustrates.

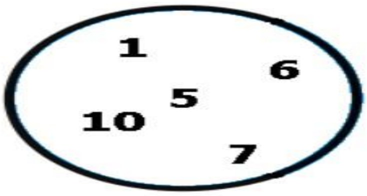


Figure 2.2: Graphical representation of the set {1, 5, 6, 7, 10}[10]

- Fuzzy logic is based on the theory of fuzzy sets, which is a generalization of classical set theory. To make a metaphor in set theory speaking, the classical set theory is a subset of the theory of fuzzy sets, as figure 2.3 illustrates.

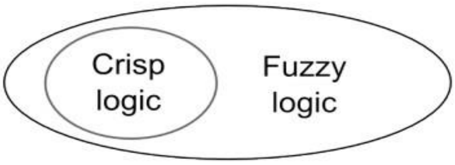


Figure 2.3:"The classical set theory is a subset of the theory of fuzzy sets"[10]

By abuse of language, following the habits of the literature, we will use the terms fuzzy sets instead of fuzzy subsets. The classical sets are also called clear sets, as opposed to vague, and by the same token classical logic is also known as Boolean logic or binary.

4 Fuzzy Logic Architecture [9]

The architecture of Fuzzy Logic consists of the following components:

4.1 Rule base

This is the set of rules along with the If-Then conditions that are used for making decisions. But, modern developments in Fuzzy Logic have reduced the number of rules in the rule base. These set of rules are also called a knowledge base.

4.2 Fuzzification

This is the step where crisp numbers are converted into fuzzy sets. A crisp set is a set of elements that have identical properties. Based on certain logic, an element can either belong to the set or not. Crisp sets are based on binary logic – Yes or No answers.

Here, the error signals and physical values are converted into a normalized fuzzy subset. In any Fuzzy Logic system, the fuzzifier separates the input signals into five states that are: Large positive, Medium positive, Small, Medium negative, Large negative.

The fuzzification process converts crisp inputs, such as room temperature, fetched by sensors and passes them to the control system for further processing. A Fuzzy Logic control system is based on Fuzzy Logic. Common household appliances, such as air-conditioners and washing machines have Fuzzy Control systems within them.

4.3 Inference Engine

The inference engine determines how much the input values and the rules match. The rules are applied based on the input values received. Then, the rules are used to develop control actions. The inference engine and the knowledge base together are called a controller in a Fuzzy Logic system.

4.4 Defuzzification

This is the inverse process of fuzzification. Here, the fuzzy values are converted into crisp values by mapping. There will be several defuzzification methods for doing this, but the best one is selected as per the input. This is a complicated process where methods, such as the maximum membership principle, weighted average method and centroid method, are used.

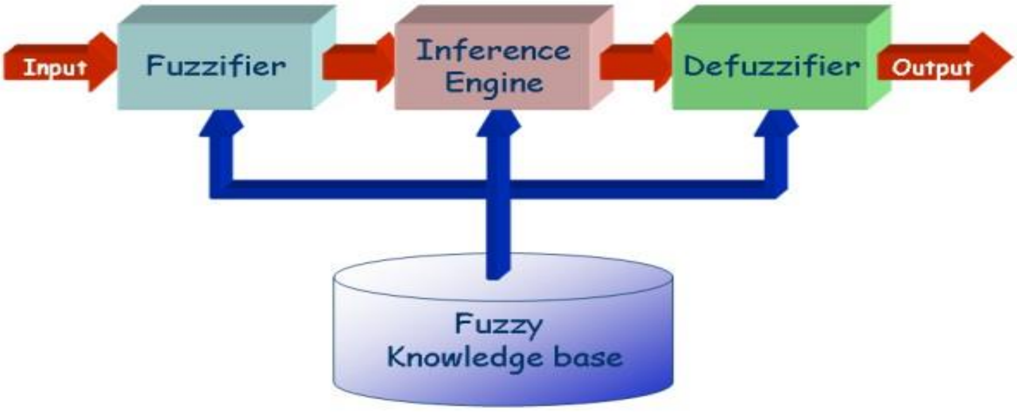


Figure 2.4: Overview diagram of a fuzzy system [10]

5 Advantages and Disadvantages of Fuzzy Logic in Artificial Intelligence [9]

Advantages	Disadvantages
- It is a robust system where no precise inputs are required - These systems are able to accommodate several types of inputs including vague, distorted or imprecise data - In case the feedback sensor stops working, you can reprogram it according to the situation - The Fuzzy Logic algorithms can be coded using less data, so they do not occupy a huge memory space - As it resembles human reasoning, these systems are able to solve complex problems where ambiguous inputs are available and take decisions accordingly - These systems are flexible and the rules can be modified - The systems have a simple structure and can be constructed easily - You can save system costs as inexpensive sensors can be accommodated by these systems	- The accuracy of these systems is compromised as the system mostly works on inaccurate data and inputs - There is no single systematic approach to solve a problem using Fuzzy Logic. As a result, many solutions arise for a particular problem, leading to confusion - Due to inaccuracy in results, they are not always widely accepted - A major drawback of Fuzzy Logic control systems is that they are completely dependent on human knowledge and expertise - You have to regularly update the rules of a Fuzzy Logic control system - These systems cannot recognize machine learning or neural networks - The systems require a lot of testing for validation and verification

Table 2.1: Advantages and Disadvantages of Fuzzy Logic in Artificial Intelligence

6 Implementation [11]

6.1 When can fuzzy rule bases be used?

Fuzzy rule bases can be chosen to solve application problems when the following conditions are satisfied:

- it is possible to act on the process (controllability),
- an expertise or knowhow exists,
- the variables (inputs and outputs) can be measured or observed, (measurability),
- qualitative expertise (if it is mathematical, conventional automatic control should be preferred),
- gradual expertise (if it is boolean, expert systems are more suitable).

6.2 Designing an application

6.2.1 Choice of operators

In most applications, "Mamdani" rule bases are used. This choice is suitable except if the expertise contains indeterminations. In most cases, the choice is also made to use "trapezoidal" membership functions as they are easier to implement and simplify the gathering of expertise. Output membership functions are often singletons, except when rules are concatenated. A triangular output membership function in fact implies an uncertainty on the output to be applied, and does not have much effect on interpolation between the rules.

Defuzzification takes place using the "centre of gravity" for control (all active rules are taken into account): the use of the "average of maxima" for decision-making problems enables a decision to be made when rules are "conflicting" and avoids intermediate decisions.

6.2.2 Methodology

Designing a fuzzy rule base is an interactive process. The largest portion of the task consists of collecting knowledge. One of the advantages of fuzzy logic is the possibility of having the rule base validated by the people who provided the expertise before testing it on a real system.

6.2.3 Collecting knowledge

This is a three-step process:

- listing the variables to be taken into account: they will become the linguistic variables of the rule base;
- listing the qualitative quantities to be taken into account and specifying when they are true and false: these quantities will become the linguistic terms of the rule base;
- formulate how these concepts are manipulated: which cases should be considered, how they are characterized, how should you act in each case.

Transcription in fuzzy rule form is then straight forward. However as few membership functions and rules as possible should be written in order to limit the number of parameters which will have to be tuned later on and to ensure legibility of the base. We observe that it is easier to add rules in order to take new situations into account than to remove them.

6.2.4 Validating the knowledge base

This takes place in a number of steps:

- presentation of the rule base to the experts who helped collect knowledge, and discussion. The aim of the discussion is to identify points that have not been covered and to ensure that the rules are understood by everyone;
- "open loop" simulation: the experts compare the behavior of the rule base to the behavior that they expect on cases chosen beforehand;
- if the process can be simulated, closed loop simulations can also be performed

6.2.5 Tuning

The rule bases written in this manner often give satisfaction right away. However the rule base may need to be modified or tuned. The following principles will act as a guideline in searching for the probable cause of the deviation observed:

- if the behavior of the closed loop controller is the opposite to what you expected, some rules have most likely been incorrectly written;
- if you wish to optimize performance, it is normally preferable to properly tune the membership functions;
- if the system is not robust and works in some cases but not all the time, it is likely that not all cases have been taken into account and that rules must be added.

6.3 Using an application

6.3.1 The function of the operators

The degree of involvement of operators controlling an application using fuzzy logic varies considerably. The following cases can be observed:

- completely autonomous system: the end-user is not familiar with fuzzy logic and is not aware of its use,
- fuzzy logic is a "black box" which can be disconnected or changed to "manual mode" by the operator,
- the operator is able to modify (tune) the membership functions according to the situation, and he does this for a production change (for example);
- the operator is able to read the rules (e.g. their degree of activation): he understands and is able to interpret the actions of the rule base. For example he can control the rule base in exceptional situations;
- the operator is the main designer of the base: he has been given the means to record his own knowhow and to validate the resulting behavior.

6.3.2 Production changes

During an application, the rule base must be able to be adapted to changes in the production system and the products manufactured. These changes can be of various kinds:

- objectives have changed (cooking temperature, etc.), for example due to a change in product manufactured. The set points or rule input membership functions must then be modified;
- system dimensions have changed; the membership functions must then be modified;
- the type of system has changed (e.g. portage of the rule base from one machine to another);the rules and membership functions must then be modified. The most common changes are of the first type. They can then be managed by qualified operators.

6.4 Choosing the implementation technology

Most of today’s applications run on standard hardware platforms (micro-controller, microprocessor, programmable controller, micro-computer, etc.). Many software programs designed to help develop fuzzy rule bases and aimed at micro-controllers, programmable controllers and micro-computers (to name but a few), enable rapid implementation of fuzzy rule bases without programming.

Fuzzy inferences can be directly programmed (assembler, C language, etc.). The disadvantage of this solution is that it is slower in the prototype phase and requires programming skills and command of fuzzy logic algorithms.

Now commonly integrated inside micro-controllers, even low cost ones, where they are used to accelerate fuzzy inferences. For applications with exacting response time demands or in order to obtain very low mass production cost prices, use of fuzzy logic ICs is advantageous. Use of such electronic chips is increasing as:

- the operations required to produce fuzzy inferences are elementary and feasible in integers.
- some operations can be carried out in parallel.
- the calculation takes place in successive steps, thereby enabling simple “pipeline” architectures to be made.

In particular, numerous ASIC components designed for specific markets exist (car, electrical household appliances, etc.). They are now commonly integrated inside micro-controllers, even low cost ones, where they are used to accelerate fuzzy inferences.

6.5 Standards

6.5.1 Components

Absence of standards is one of the main problems holding up the use of fuzzy logic chips. This is because these components are not compatible with one another as each one is the result of choices made by manufacturers.

6.5.2 Software

Regarding software, lack of portability has also slowed down widespread use of fuzzy logic in industry. Today, a work group in which Schneider plays an active part, has incorporated the "fuzzy logic" language standard into the language standard of programmable controllers (first official draft of standard IEC 61131-7 available in 1997). Other initiatives in the field of fuzzy logic standardization should spring from this.

7 Applications of Fuzzy Logic [9]

The applications of Fuzzy Logic are spread across several fields. They are as follows:

- **Medicine**
  - Controlling arterial pressure when providing anesthesia to patients.
  - Used in diagnostic radiology and diagnostic support systems.
  - Diagnosis of prostate cancer and diabetes.
- **Transportation systems**
  - Handling underground train operations.
  - Controlling train schedules.
  - Braking and stopping vehicles based on parameters, such as car speed, acceleration and wheel speed.
- **Defense**
  - Locating and recognizing targets underwater
  - Supports naval decision making
  - Using thermal infrared images for target recognition
  - Used for controlling hypervelocity interceptors
- **Industry**
  - Controlling water purification plants
  - Handling problems in constraint satisfaction in structural design
  - Pattern analysis for quality assurance
  - Fuzzy Logic is used for tackling sludge wastewater treatment
- **Naval control**
  - Steer ships properly
  - Selecting the optimal or best possible routes for reaching a destination
  - Autopilot is based on Fuzzy Logic
  - Autonomous underwater vehicles are controlled using Fuzzy Logic

8 Conclusion

Although Fuzzy Logic in artificial intelligence helps to mimic human reasoning, these systems need expert guidance to be built. This lets you rely on the experience of experts who have a better understanding of the system. Fuzzy Logic can also be used for enhancing the execution of algorithms. IBM Watson uses Fuzzy Logic and fuzzy semantics. In the next chapter we're going to introduce the fuzzy ontologies.

Chapter III

Fuzzy Ontologies

1 Introduction

Conceptual formalism supported by typical ontologies may not be sufficient to represent uncertainty information which is caused due to the lack of clear cut boundaries between concepts of a domain. Fuzzy ontologies are proposed to offer a way to deal with this uncertainty. Indeed they were proposed as a combination of fuzzy theory and ontologies to tackle such problems. What are Fuzzy ontologies? why they are needed? how they are represented?…etc.

In this chapter we're going to answer all of these questions.

2 The definition of Fuzzy Ontology

Definition 1: [12]

- Crisp ontology is a 4-tuple O= (C, P, R and A), where:
    1. C is a set of concepts defined for the domain. A concept is often considered as a class in ontology.
    2. P is a set of concept properties.
    3. R is a set of binary semantic relations defined between concepts in C.
    4. A is a set of axioms. An axiom is a real fact or reasoning rule.

- Fuzzy ontology:

Fuzzy ontologies are an extension of the domain of crisp ontologies for solving the uncertainty problems. Current fuzzy ontological models do not focus on essential semantic relationships between fuzzy concepts, which lead difficulty in ontology integrating. To represent formally the fuzzy knowledge, we apply the fuzzy logic and we propose an appropriate model.

Definition 2: [13]

A fuzzy ontology is a quintuple F =< I, C,T,N,X > where:

- I is the set of individuals (objects), also called instances of the concepts.
- C is a set of fuzzy concepts (or classes - cf. in OWL - of individuals, or categories, or types). Each concept is a fuzzy set on the domain of instances.
- The set of entities of the fuzzy ontology is defined by E = C ∪ I.
- T denotes the fuzzy taxonomy relations among the set of concepts C. It organizes concepts into sub-(super-)concept tree structures. The taxonomic relationship T (i, j) indicates that the child j is a conceptual specification of the parent i with a certain degree.
- N denotes the set of non-taxonomy fuzzy associative relationships that relate entities across tree structures, for example: - Naming relationships, describing the names of concepts - Locating relationships, describing the relative location of concepts - Functional relationships, describing the functions (or properties) of concepts
- X is the set of axioms expressed in a proper logical language, i.e., predicates that constrain the meaning of concepts, individuals, relationships and functions.

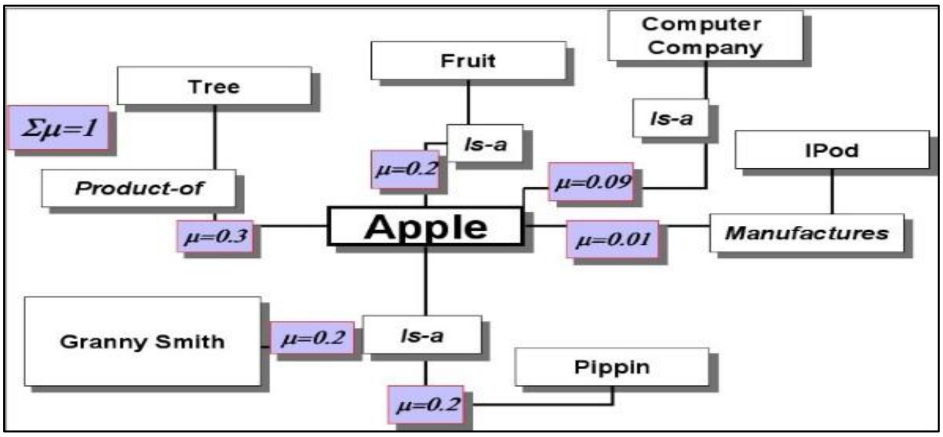


Figure 3.1: fuzzy ontology scheme.[13]

Example:[13]

A fuzzy ontology for wine:

- **Illustration of a non-taxonomic relation:**

✓ Name of fuzzy relation: Taste.

✓ Name of instance: cabernet.

✓ Name of property: dry.

- **Some possible values:**

✓ Taste(cabernet, dry)=0.7⇔ Cabernet has a dry taste with value 0.7 .

✓ Taste(cabernet, very_dry)=0.7×0.7 ⇔ Cabernet has a very dry taste with value 0.49.

✓ Taste(cabernet, not_dry)=1-0.7 ⇔ Cabernet has not a dry taste with value 0.3.

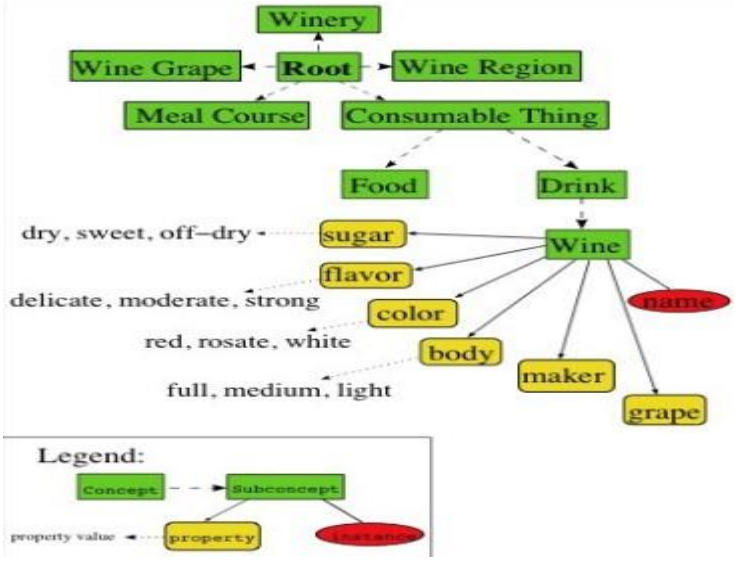


Figure 3.2: fuzzy ontology for wine[13]

3 Why Fuzzy Ontology Is Needed?

Classical Ontologies have some limitations for representing knowledge in real world. Firstly, concepts which are used to describe real world in some domain are inherently vague. For example, a town is a small and less crowded city, but small and crowded are not clearly well defined. It may be difficult to say that a special region fully belongs to the category of towns or cities. It is more desirable to state that it belongs to both of these concepts with different degrees. Similarly, for having a concept hierarchy it is more rational to have a fuzzy concept hierarchy where a town can be considered as a sub-concept of a city to some degree. Furthermore, a concept may have attribute values taking vague values, for example a person can have attribute “age” as a linguistic variable taking values “young”, “middle aged” or “old” defined by means of fuzzy numbers. And finally, in some domains we may need fuzzy relations between entities. For example, in the domain of Molecular Biology a relation can be associated with different biological entities with different degrees of strength. In the work of Abulaish and Dey it is observed that the relation “inhibitor of” occurs frequently between protein molecule and protein complexes while the same relation is rare between two protein molecules. In addition to the inherent uncertainty that may happen in some real-world domains, uncertainty may happen due to the nature of application. For example in a search engine in the case of overloaded concepts1, the location of concept may be different for users with different preferences. Also, the uncertainty may be because of differences in experts’ conceptualization of a domain. For example, New York Times categorizes news into “World, Us, NY/Region, Business, Technology, Science, Health, Sports, Opinion, Arts, Style and Travel”, While CNN categorizes news into categories “Home, World, US, Politics, Entertainment, Health, Technology, Travel, Living, Business, Sport and Time”. A one-to-one mapping which may be required by an application like news summarization is not possible without inferring a kind of uncertainty [14].

4 How Fuzzy Ontology Is Represented? [14]

As the theoretical counterpart of fuzzy ontology, the fuzzy description logics (FDLs) have also attracted much attention from researchers. Concepts and relations in fuzzy description logic are fuzzy sets. For example, the concept “YoungMan” is a fuzzy concept and is defined as:

$$\text{YoungMan}=\text{Man}\cap\exists\ \text{Age.Young}$$

“age” expression is a fuzzy concept (linguistic variable) and it gets fuzzy linguistic value:” young”. Fuzzy linguistic value “young” may be defined with a trapezoidal function such as:

$$\mu_{Young}(c.Age)=\begin{cases} 1 & if B \leq C.Age \leq C \\ \frac{D-C.Age}{D-C} & if C < C.Age < D \\ \frac{C.Age-A}{B-A} & if A < C.Age < B \\ 0 & Otherwise \end{cases}$$

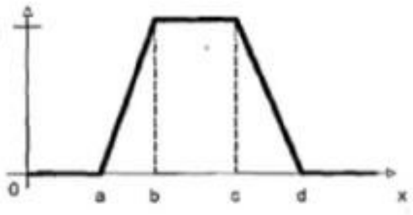


Figure 3.3: Fuzzy trapezoidal function for concept "YoungMan".[14]

5 Fuzzy Ontologies Formal Description [12]

The definition of the basic components of a fuzzy ontological model is very delicate. Indeed, it must be complete and not redundant. The integration of fuzzy logic in the description logic is done principally by the definition of membership functions associated with concepts and relations. For example, the concept "Young Person" is defined based on the membership function associated with the linguistic variable "Young". The degree of membership of element to the concept "Young Person" is given by the following trapezoidal formula:

$$\text{Trz (Age;12,18,30,40)}=\begin{cases} (\text{Age}-12)/(18-12) \;\; if \;\; 12 \leq Age \leq 18 \\ 1 \;\; if \;\; 18 < Age \;\leq 30 \\ (-\text{Age}+40)/(40-30) \;\; if \;\; 30 < Age \leq 40 \\ 0 \;\;\; if \;\; Age < 12 \; Or \; Age > 40 \end{cases}$$

We do not use the expression of "Young Person". We rather propose the following expression which appears mathematically more rigorous to us:

Young-Person = Person ∩Young (Age; 12, 18, 30, 40; s)

We equally adopt the some idea to the definition of fuzzy ontological components. We hereafter define the various components of our fuzzy ontological model. Whatever the used ontology language, the description of an ontology is based on the concept of classes, relations and properties. This description is based on an exact logic and cannot be used for the description of fuzzy ontologies. We then propose to define a new type of class, entitled "Fuzzy_Class", and a series of the types of classes which specialize the first (cf. Figure 3.4). These types are intended for the design of fuzzy ontologies:

- **Fuzzy_Class:** it is the root class which represents a generalization of various fuzzy classes. It has "Degree_of_Membership" property, which is a calculable property.
- **Fuzzy_Class_Trz**: it is a specialization of the class "Fuzzy_Class". It is used to represent fuzzy concepts which have trapezoidal functions. It has 4 properties Fuzzy_a, Fuzzy_b, Fuzzy_c and Fuzzy_d which represents respectively the attributes "a", "b", "c" and "d" of the corresponding membership function.
- **Fuzzy_Class_Tri:** it is a specialization of the class "Fuzzy_Class". It is used to represent fuzzy concepts which have triangular functions. It has 3 properties Fuzzy_a, Fuzzy_b, and Fuzzy_c which represent respectively the attributes "a", "b" and "c" of the corresponding membership function.

- **Fuzzy_Class_L:** it is a specialization of the class "Fuzzy_Class". It is used to represent fuzzy concepts which have L-functions. It has 2 properties Fuzzy_a and Fuzzy_b to represent respectively the attributes "a" and "b" of the corresponding membership function.
- **Fuzzy_Class_R:** it is a specialization of the class "Fuzzy_Class". It is used to represent fuzzy concepts which have R-functions. It has 2 properties Fuzzy_a and Fuzzy_b to represent respectively the attributes "a" and "b" of the corresponding membership function.

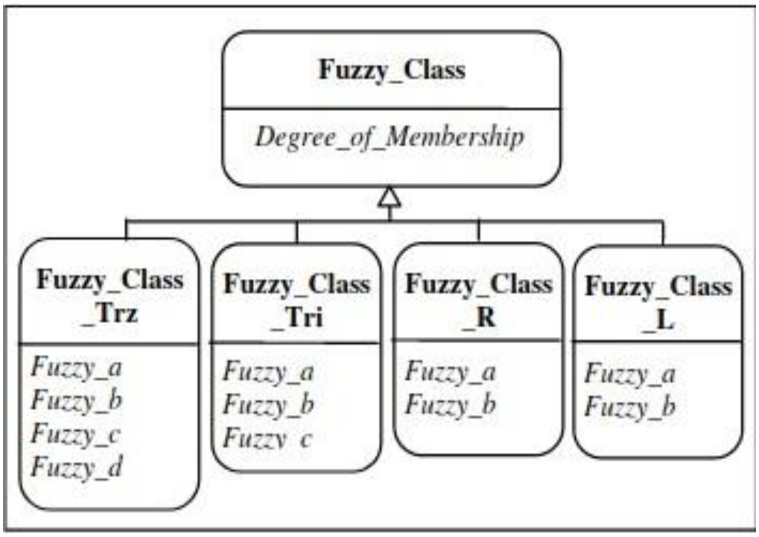


Figure 3.4: Fuzzy ontological components structuring [12]

6 Conclusion

In this chapter, we have presented the fuzzy ontology as a good solution for the uncertainty problems, its definition, the need behind it, its representation and its formal Description. In the next chapter we're going to introduce our proposed medical diagnostic help system using fuzzy ontologies

Chapter IV

The Proposed Medical Diagnostic Help System Using Fuzzy Ontologies

1 Introduction

Among the goals of ontology is to help designing the real world with all semantic restrictions, but there are some fuzzy concepts that cannot be designed using classical ontologies. Fuzzy ontology can solve the problem of the representation of fuzzy concepts found in many fields. In this chapter, we will address the problem presented in this report, To do this, we will create a fuzzy ontology using the protege editor and the Fuzzy OWL2 plugin to add the fuzzy part of the ontology, at the end, we will send requests to test our ontology and we will program a program that manages this ontology.

2 Choice of the construction method of our ontology

To build our ontology, we have to choose the construction method. In this work, we chose the "METHONTOLOGY" method to build our ontology because it is about one of the most famous methods used in the construction of ontologies. This is the most important step in the process of building ontology. It is inspired by the "METHONTOLOGY" methodology which consists of identifying and structuring, from sources of information, domain knowledge. They make it possible to arrive at a set of semi-formal intermediate representations independently of the languages of formalizations to be used to represent the ontology "METHONTOLOGY" [20].

3 Stages of the construction process of our ontology

The construction of our ontology takes place in five stages and it is presented as follows:

- Needs assessment (Needs specification)
- Conceptualization
- Formalization
- Implementation
- Verification.

3.1 Needs assessment

The area of medical diagnostics is one of the areas that require special attention as it is a sensitive area requiring precision, but this area includes fuzzy concepts and relationships that are difficult to manage.

In this work, we define and create a fuzzy ontology that makes it possible to represent and manipulate the knowledge and data relating to the diagnosis of pneumonia disease and influenza, while taking into consideration the fuzzy aspects (imprecision and uncertainties).

The flu, or influenza, is a disease caused by influenza viruses of the Orthomyxoviridae family, RNA viruses. Contagious disease, influenza first affects the system respiratory system and may become complicated or have severe forms [18].

Pneumonia is an infection of the lungs most often caused by a virus or bacteria. The infection more specifically affects the pulmonary alveoli, these tiny bags made of shaped like balloons located at the ends of the bronchioles. It usually affects only one of the 5 lobes of the lung (3 lobes in the right lung and 2 in the left), hence the term pneumonia lobar. When pneumonia also affects the bronchi, it is called bronchopneumonia [17].

Covid-19 refers to "Coronavirus Disease 2019", the disease caused by a virus of the Coronaviridae family, SARS-CoV-2, it is a respiratory disease that can be fatal in patients weakened by age or other chronic disease.

The main symptoms of the disease are fever, fatigue and a dry cough. Some patients also experienced pain, stuffy and runny nose, sore throat and diarrhea. These symptoms are usually mild. But about one in six people have more severe symptoms, including dyspnea.[19]

The reason why we have chosen these three types of illnesses is to contain vague concepts in their symptoms, such as fever is an increase in temperature. Body it can be high or very high there is in this case a kind of ambiguity due to the overlap of these concepts, it is difficult to determine the degree of these concepts. The fuzzy ontology based on fuzzy logic solves such problems, for example, the patient concept is identified as a fuzzy class according to the linguistic variable "the temperature of the body". It is important to know the temperature of the patient's body for a more reliable diagnosis and management. The patient class is fuzzified into three fuzzy classes which are: fever with an increasing monotonic membership function which has as parameters a = 37 and b = 40, normal temperature with a trapezoidal membership function which has as parameters a = 35, b = 36, c = 37 and d = 40, hypothermia with decreasing monotone membership function which has parameters a = 35 and b = 36.

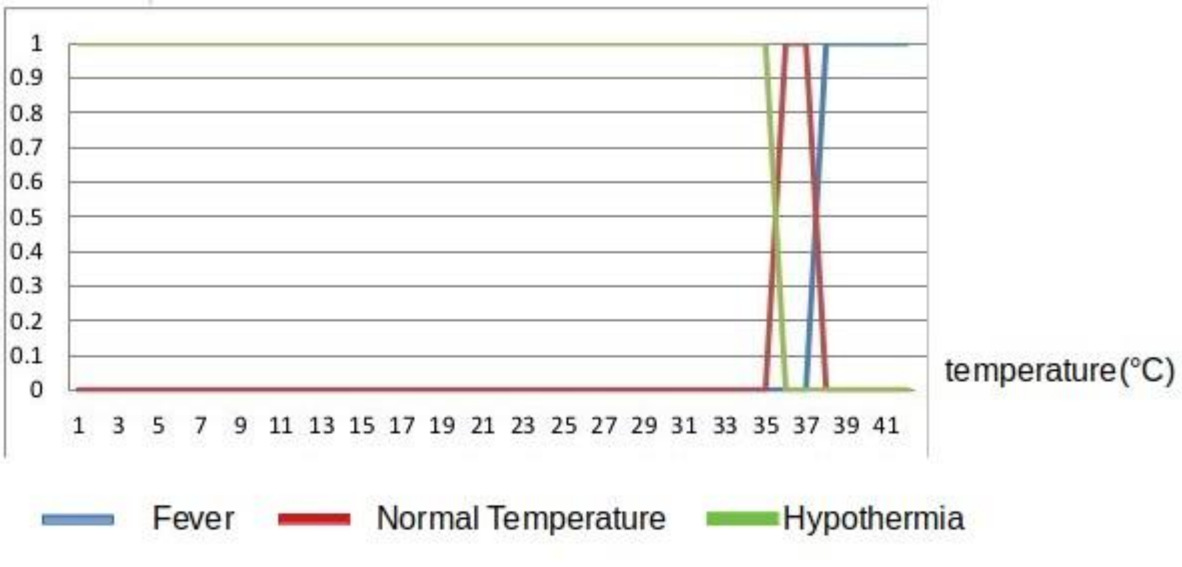


Figure 4.1: The membership functions of the fuzzy classes of the Patient concept.

For example, the patient whose temperature is 36 ° C, his body temperature is said to be low (Hypothermia) with a membership degree of 0.5. It is also said that his body temperature is moderate (normal temperature) with a membership degree equal to 0.5. But in this case his body temperature is not high (Fever). To make our ontology easy to use, we need to program a system that handles this ontology. The following UML Diagrams describe the main elements of our system.

3.1.1 Use case diagram

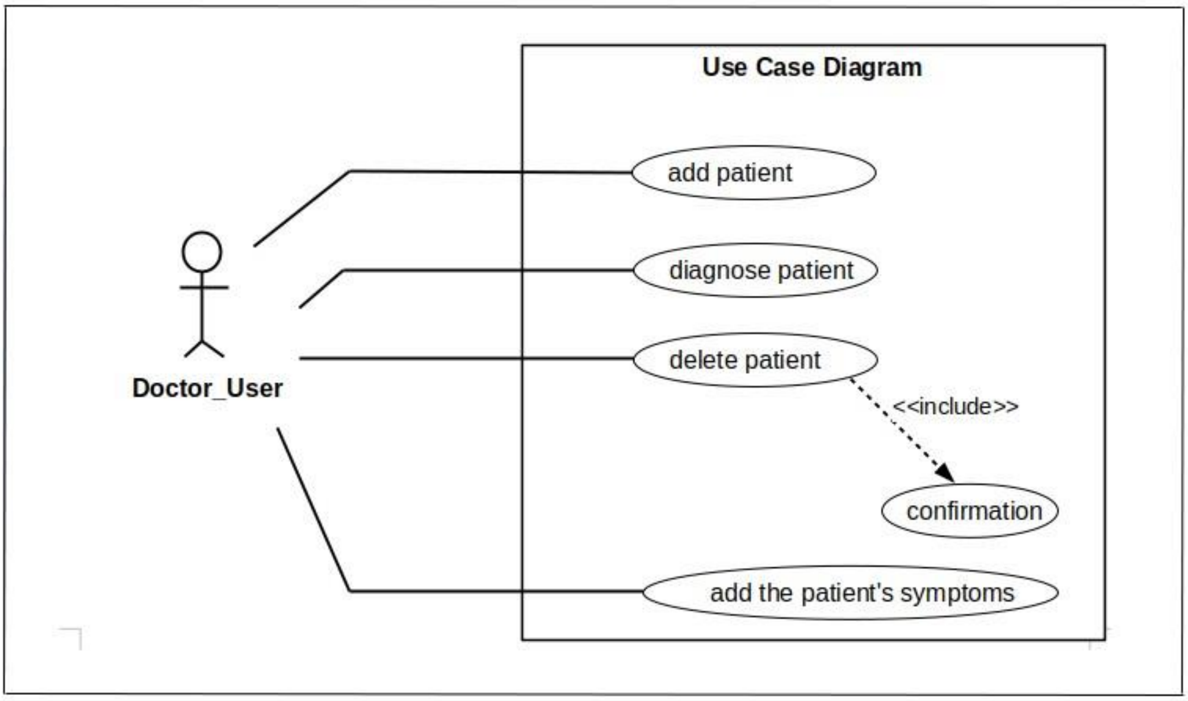


Figure 4.2: Use case diagram.

3.1.2 Add patient sequence diagram

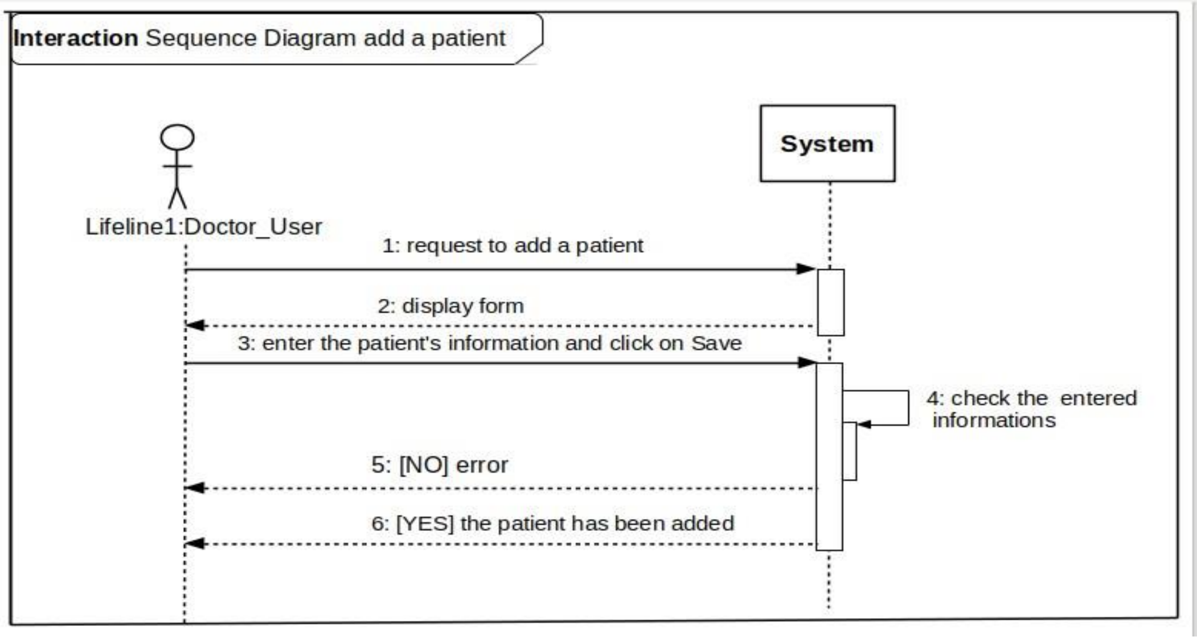

Interaction Sequence Diagram add a patient
System
Lifeline1:Doctor_User
1: request to add a patient
2: display form
3: enter the patient's information and click on Save
4: check the entered informations
5: [NO] error
6: [YES] the patient has been added

Figure 4.3: Add a patient.

3.1.3 Class diagram

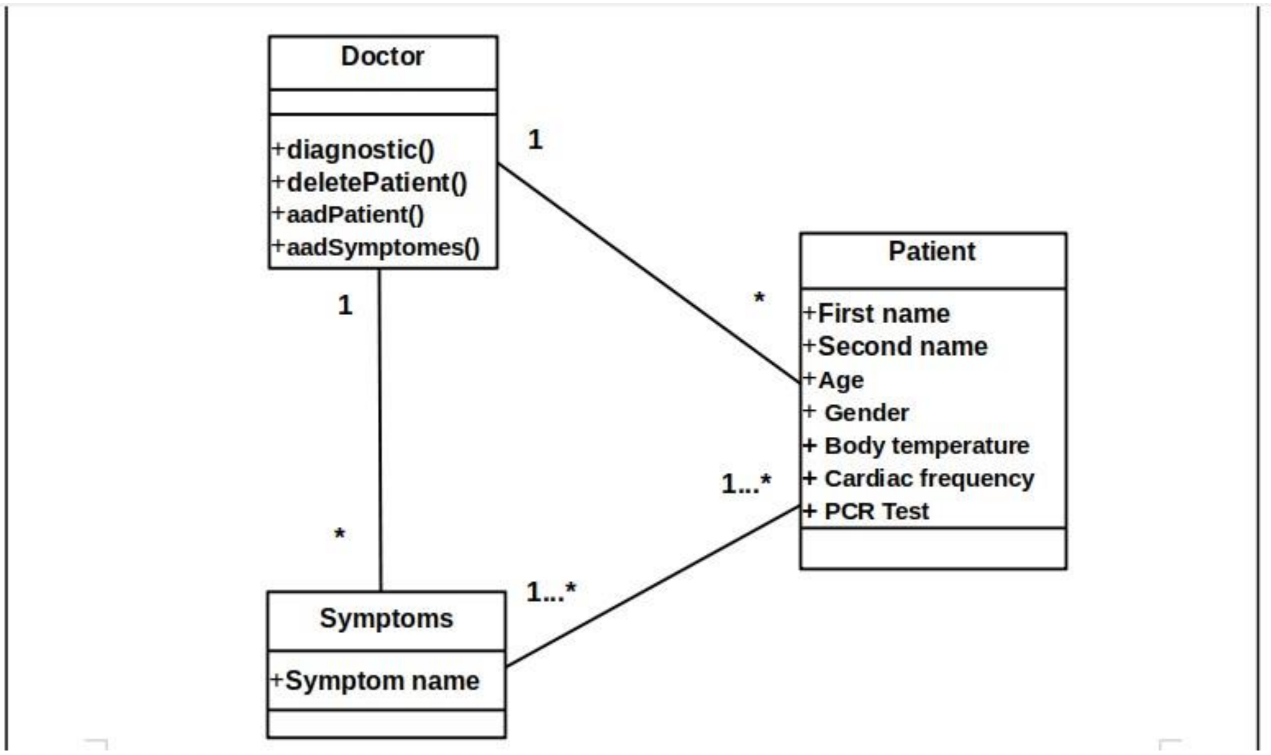


Figure 4.4: Class diagram.

3.2 Conceptualization

The conceptualization step is one of the most important steps because it gives an accurate description of the ontology. The purpose of this step is to organize and structure the knowledge acquired during the knowledge acquisition activity. For this we distinguish the following main activities:

- Construction of the glossary of terms
- Construction of the taxonomy of concepts
- Construction of the dictionary of precise concepts
- Construction of the table of binary relations
- Construction of the instance attributes table
- Construction of the table of class attributes
- Table of instances of precise concepts
- Table of fuzzy assertions
- Construction of the fuzzy part

3.2.1 Construction of the glossary of terms

The glossary is a collection and a description of all the terms that can be used in the field to be studied or dealt with. The table below includes a set of related terms to the field of medical diagnosis with their descriptions organized in the form of a glossary.

Term	**Description**	**Type**
Medical help	Is the care and assistance provided by the doctor to the patient for treatment.	Class
Patient	The patient is anyone who receives care medical or medical treatment.	Class
Symptoms	A symptom is a sign, an observable disorder by a patient, which is the manifestation of a disease and its evolution.	Class
Complications	Is an unwanted development of a disease may become more prevalent in the body or sometimes increase its effects and may result in another illness due to previous illness.	Class
Disease	The disease is an impairment of the functions or health of a living organism, animal or plant	Class
Flu	Influenza is a disease caused by influenza viruses of the Orthomyxoviridae family, RNA viruses.	Sub Class
Pneumonia	Is an infection of the lungs most often caused by a virus or bacteria. The infection more specifically affects the pulmonary alveoli, these tiny bags made of shaped like balloons located at the ends of the bronchioles.	Sub Class
Covid-19	Coronavirus Disease 2019, the disease caused by a virus of the Coronaviridae family, SARS-CoV-2, it is a respiratory disease that can be fatal in patients weakened by age or other chronic disease.	Sub Class
Diagnostic	Is the process of identifying a disease or of a condition that explains the symptoms and signs of somebody.	Sub Class
Treatment	Medical treatment is an attempt to control and eliminate the patient's disease.	Sub Class
…………………	…………………………….	………………….

Table 4. 1: Glossary of terms.

3.2.2 Construction of the taxonomy of concepts

Taxonomy is a hierarchical method of constructing ontology concepts, from key concepts to secondary concepts, etc. For example, the concept of patient is a subclass of the person class, because each patient is a person. To organize concepts in the form of taxonomy, Figure 4.5 shows the hierarchy of concepts.

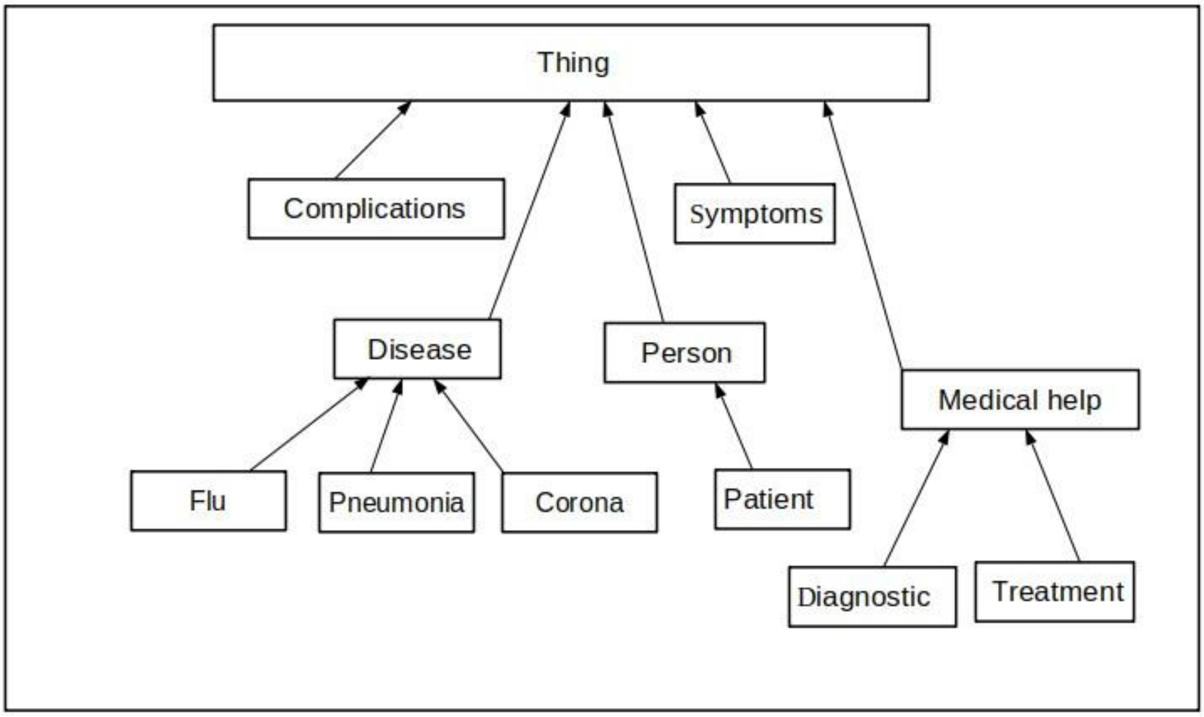


Figure 4.5: Hierarchy of concepts.

3.2.3 Construction of the dictionary of precise concepts

A precise concept dictionary contains all non-fuzzy concepts related to the field of study with their attributes and Instances. The following table shows the dictionary of precise concepts.

Concept	**Attributes**	**Instances**
Patient	-First Name -Second Name -Age -Gender	- Manal - Mourad
Disease	-Disease Name	- Grippe
Symptoms	-Symptom Name	- Fever - Cough - sore throat - Fatigue runny or stuffy nose. - Muscle or body aches. - Headaches - Fatigue - Vomiting
………………	………………	………………

Table 4.2: Dictionary of precise concepts.

3.2.4 Construction of the table of binary relations (Précised)

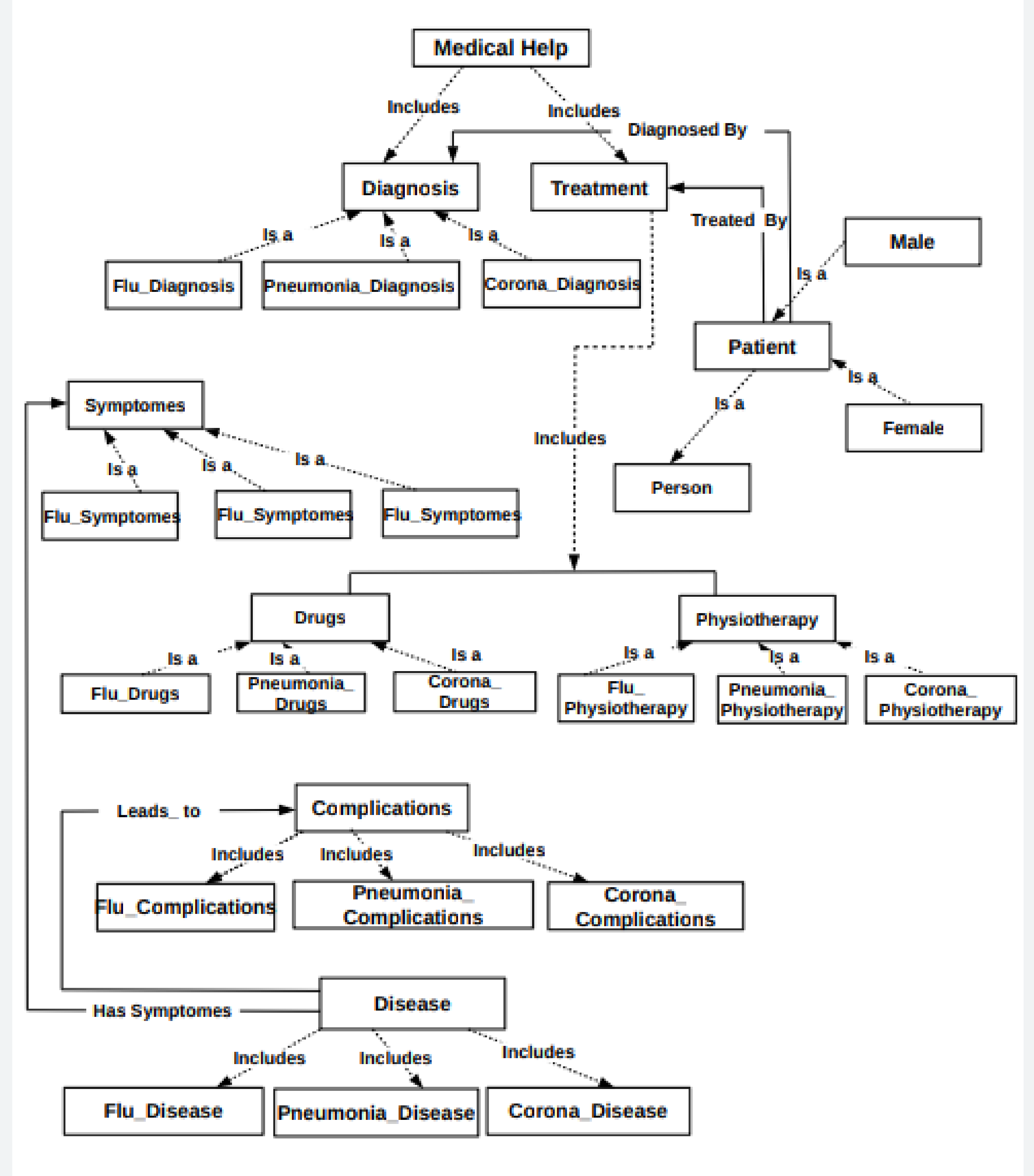


Figure 4.6: Diagram of binary relations.

3.2.5 Construction of the instance attributes table

Instance	**Concept**	**Attributes**	**Type**
Patient 1	Patient	- First Name - Second Name - Age - Gender	String String Double String
Patient 2	Patient	- First Name - Second Name - Age - Gender	String String Double String
Tachycardia	- Symptoms of pneumonia - symptoms of Covid-19	-Cardiac frequency	Double
Fever	- symptoms of Flu - Symptoms of pneumonia - symptoms of Covid-19	-Body temperature	Double
Covid-19	Disease	- PCR Test -Body temperature	Boolean
………………..	………………..	………………..	………………..

Table 4.3: Table of instance attributes.

3.2.6 Construction of the table of class attributes

Attribute	**Concept**	**Type**	**Cardinality (min/max)**	**Values domain**
First Name	Patient	String	(1,1)	-
Second Name	Patient	String	(1,1)	-
Age	Patient	Double	(1,1)	-
Gender	Patient	String	(1,1)	(Male Female)
Body temperature	Patient	Double	(1,1)	-
Cardiac frequency	Patient	Double	(1,1)	-
Antibody level	Patient	Double	(1,1)	-
PCR Test	Patient	Boolean	(1,1)	-
…………..	…………..	…………..	…………..	…………..

Table 4.4: Table of class attributes.

3.2.7 Table of instances of précised concepts

Concept	**Instance Name**	**Attributes**	**Values**
Patient	Patient 1	- First Name - Second Name - Age - Gender	Manal Ben Ali 28 Female
Disease	Disease2	Disease Name	Flu
Symptom	Symptom1	Symptom Name	Fever
……………...	……………...	……………...	……………...

Table 4.5: Instances of precised concepts.

3.2.8 Table of fuzzy assertions

Relation	**Source instance**	**Target instance**	**Degree of belonging**
has_symptomes	Ali	40°C	1
has_symptomes	Mohamed	200 beats/min	1
……………….	……………….	……………….	……………….

Table 4.6: Table of fuzzy assertions.

3.2.9 Construction of the fuzzy part

In order to make our ontology fuzzy, we need to add the fuzzy part, to do this we create the fuzzy data types, besides adding the explanatory annotations to define the fuzzy concepts. The following table represents the fuzzy data types with their descriptions and annotations.

Data Type	Data type description	Annotations
fever	Fever is a body temperature abnormally high.	<fuzzyOwl2 fuzzyType="datatype"> <Datatype type="rightshoulder" a="37.5" b="40" /> </fuzzyOwl2>
hypothermia	Hypothermia is the body temperature drops below 35 ° C.	<fuzzyOwl2 fuzzyType="datatype"> <Datatype type="leftshoulder" a="35.0" b="36.5" /> </fuzzyOwl2>
High rythme	Is increased heart rate, that is, the number of heartbeats per minute exceeds the normal number.	<fuzzyOwl2 fuzzyType="datatype"> <Datatype type="rightshoulder" a="100" b="200" /> </fuzzyOwl2>
Low rythme	Is a decrease in heart rate, that is, the number of heartbeats per minute is abnormal.	<fuzzyOwl2 fuzzyType="datatype"> <Datatype type="leftshoulder" a="50" b="60" /> </fuzzyOwl2>
Moderate rythme	Indicates that the number of heart rhythms is at its normal level.	<fuzzyOwl2 fuzzyType="datatype"> <Datatype type="trapezoidal" a="50.0" b="60.0" c="100.0" d="200.0" /> </fuzzyOwl2>
Normal_temperature	Signifies that the body temperature is at the normal level.	<fuzzyOwl2 fuzzyType="datatype"> <Datatype type="trapezoidal" a="36.5" b="36.5" c="37.5" d="40.0" /> </fuzzyOwl2>
…………………………….	…………………………….	…………………………….

Table 4.7: Types of fuzzy data.

3.3 Formalization

The formalization facilitates the representation of the ontology in an official and operational language.

3.3.1 Representation of the terminology part (T-Box)

Is represented in definitions of concepts and roles (Precise and fuzzy) and inclusions of concepts, using the syntax of the logic of SHIQ descriptions.

3.3.1.1 Definition of concepts

Concept name	**The logic of descriptions**
Patient	(∀Is a.Person) ⊓ (∃ have.disease) (∃ Diagnosed _by.Diagnostic) ⊓ (∃ Treated_by.Treatment).
Medical help	(∀Includes. Diagnosis) ⊓ (∀Incluse.Treatment).
Disease	(∃A_risck.Person) ⊓ (∃Leads_to.Complication) ⊓ (∃ has_Symptoms.Symptoms).
Symptoms	Flu_Symptoms ⊓ Pneumonia_Symptoms ⊓ Covid-19_Symptoms
Treatment	(∀Includes.Drugs) ⊓ ∀Includes.physiotherapy).
Diagnostic	Medical help ⊓ Flu_diagnosis ⊓ Pneumonia_diagnosis ⊓ Corona_diagnosis
………………………	…………………………….

Table 4.8: Definition of precise concepts.

3.3.1.2 Definition of relations

Has_Disease(Patient,Disease)
Treated_by (Patient, Treatment)
Diagnosed_by (Patient, Diagnosis)
Has_ Symptoms (Disease, Symptoms)
Leads_to (Disease, complication)
.... ……

Table 4.9: Definition of relations.

3.3.1.3 Inclusion of concepts

Medical help ⊆ Thing
Complications ⊆ Thing
Disease ⊆ Thing
Person ⊆ Thing
Symptoms ⊆ Thing
Diagnosis ⊆ Medical Help
Treatment ⊆ Medical Help
Diagnosis_Pneumonia ⊆ Diagnosis
Diagnosis_Flu ⊆ Diagnosis
Diagnosis_Covid-19 ⊆ Diagnosis
Drugs ⊆ Treatment
Physiotherapy ⊆ Treatment

Complications _Pneumonia ⊆ Complications
Complications _Flu ⊆ Complications
Complications _Covid-19 ⊆ Complications
Disease_Pneumonia ⊆ Disease
Disease_Flu ⊆ Disease
Disease_Covid-19 ⊆ Disease
Patient ⊆ Person
Male ⊆ patient
Female ⊆ patient
Symptoms_Flu ⊆ Symptoms
Symptoms_Pneumonia ⊆ Symptoms
Symptoms_Covid-19 ⊆ Symptoms

Table 4.10: Inclusion of concepts.

3.3.2 Representation of the assertional part (A-Box)

A-Box is a set of axioms describing concrete situations.

Mourad: Patient
38.0 ° C: Fever
100 beats / min: Heart rate
Male: Gender
PCR Test: negative
Diagnosis: Boukhalfa Ahmed
Treatment: Makhloufi Ali
....

Table 4.11: The assertional part (A-Box).

4 Conclusion

In this chapter, we have presented in detail the steps in which we created our ontology and explained our work by many diagrams. In the next chapter we will explain how our system can be implemented.

Chapter V

The Implementation

1 Introduction

The implementation of the fuzzy ontology is the same as the implementation of the classical ontology the difference is to add the fuzzy part only, so to implement our ontology we used the protégé editor version 5.5.0, Fuzzy OWL2 plug-in and language OWL offered by protégé and finally Fuzzy DL reasoner for the verification.

2 The Used Programming Tools

2.1 Protégé

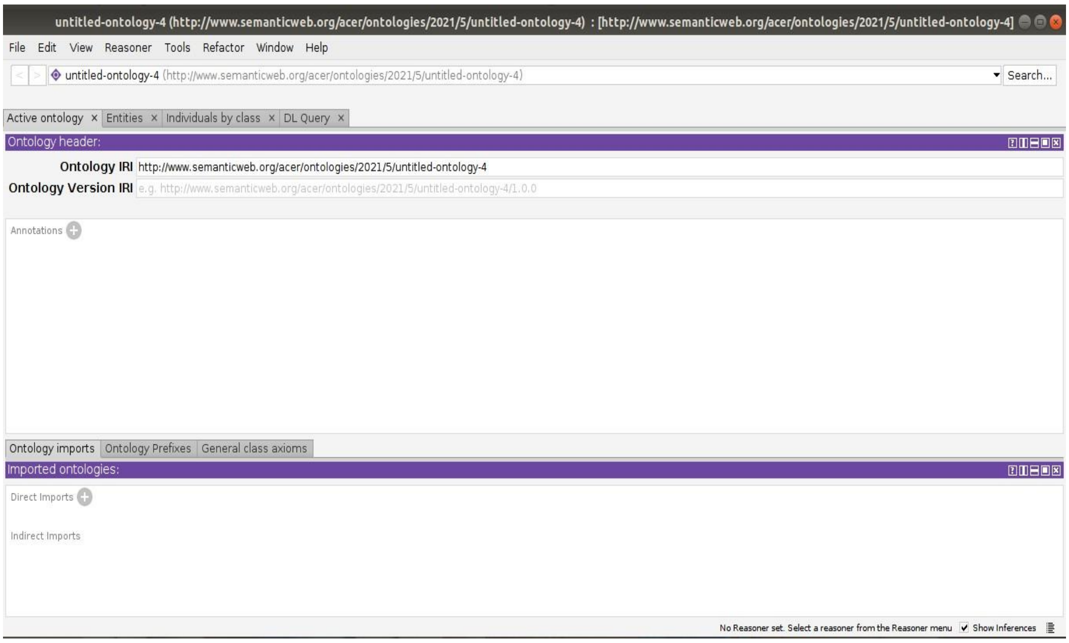
2.1.1 Definition

Protégé is an open-source ontology editor that supports OWL (Ontology Web Language). It is available for free download from the protégé site.

Figure 5.1: Protégé logo

2.1.2 Installation Process

The installation of Protégé 5.5.0 is very simple just extract the loaded package from the protégé site to the wanted location, double click on the extracted folder then double click on the executable file. The following figure shows the main interface of the protégé editor 5.5.0 after installation.



untitled-ontology-4 (http://www.semanticweb.org/acer/ontologies/2021/5/untitled-ontology-4) : [http://www.semanticweb.org/acer/ontologies/2021/5/untitled-ontology-4]

untitled-ontology-4 (http://www.semanticweb.org/acer/ontologies/2021/5/untitled-ontology-4)

Ontology header:

Ontology IRI http://www.semanticweb.org/acer/ontologies/2021/5/untitled-ontology-4

Ontology Version IRI e.g. http://www.semanticweb.org/acer/ontologies/2021/5/untitled-ontology-4/1.0.0

Imported ontologies:

No Reasoner set. Select a reasoner from the Reasoner menu ☑ Show Inferences

Figure 5.2: Protégé 5.5.0 graphical interface.

2.2 Fuzzy Owl2 Pluging

2.2.1 Definition

Fuzzy OWL 2 is a plugin for the Protégé editor that allows users to edit, save Fuzzy OWL 2 ontologies and submit queries to the FuzzyDL inference engine. It was developed by Fernando Bobillo and Umberto Straccia in 2011.

2.2.2 Installation Process

We copy the "FuzzyOWL.jar" file into the protected "plugins" file and we copy the "dlib" file into the protected file. The following figure shows the plugin's fuzzy owl2 tab after installation.

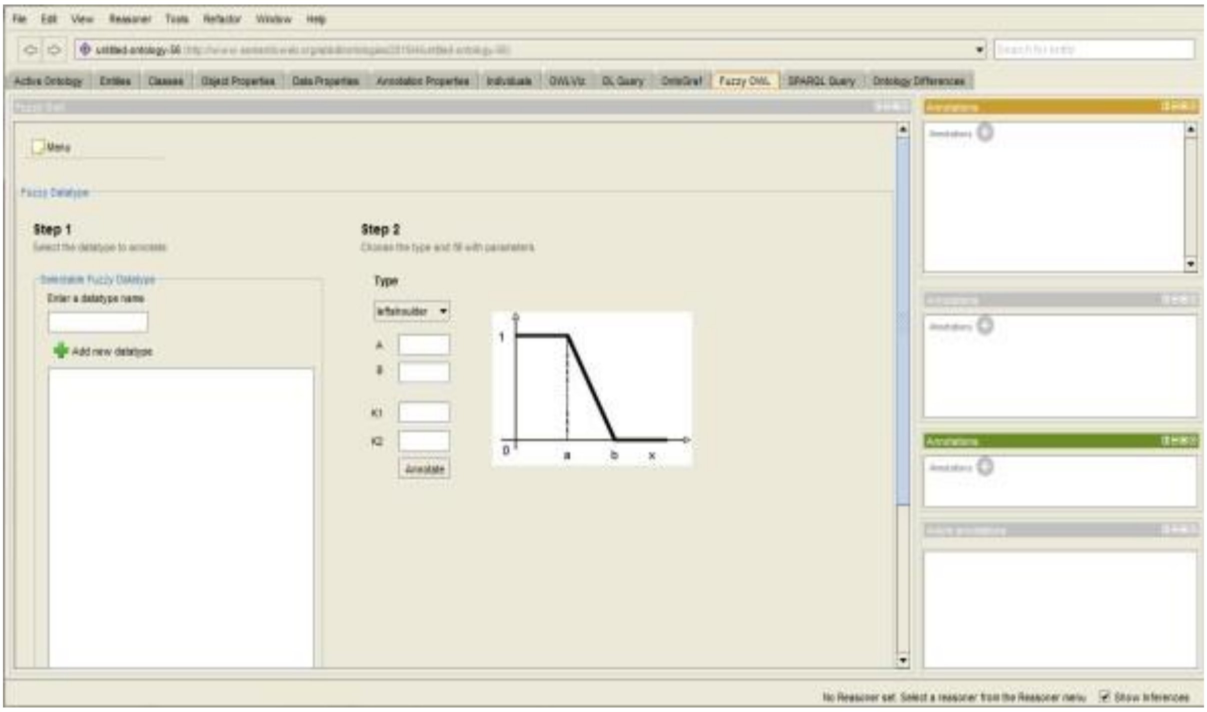
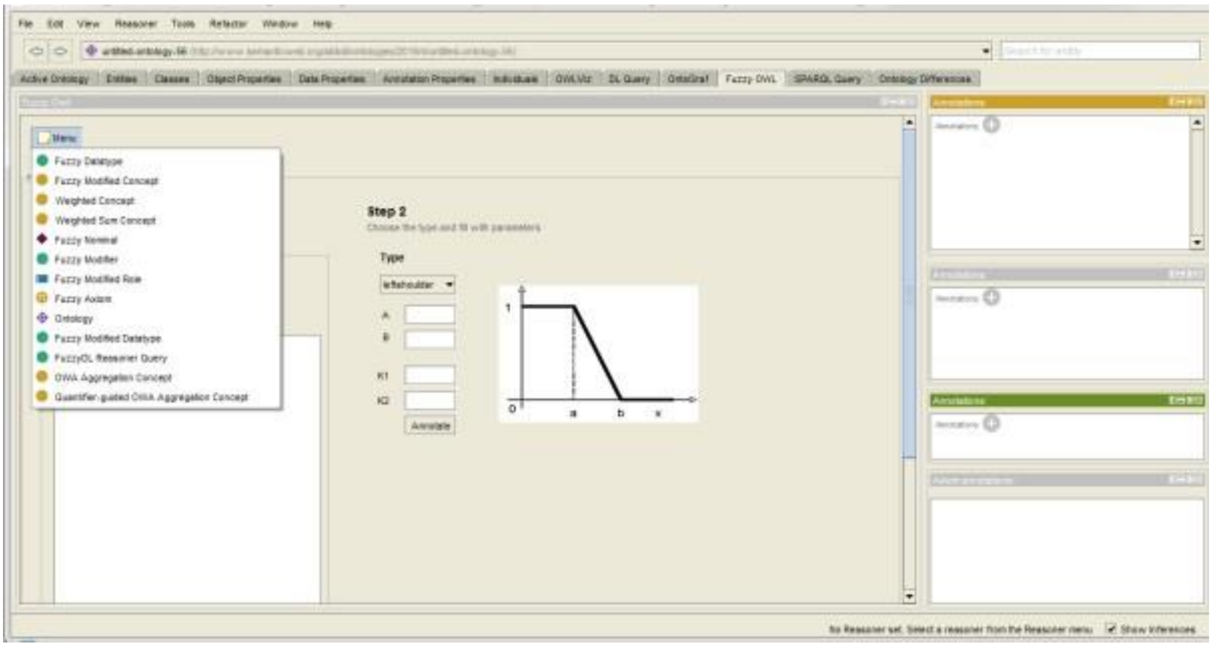


Figure 5.3: The plugin's fuzzy owl2 tab

The Fuzzy OWL 2 plugin has a menu of choices that we need to include the fuzzy part of the ontology.



Step 2

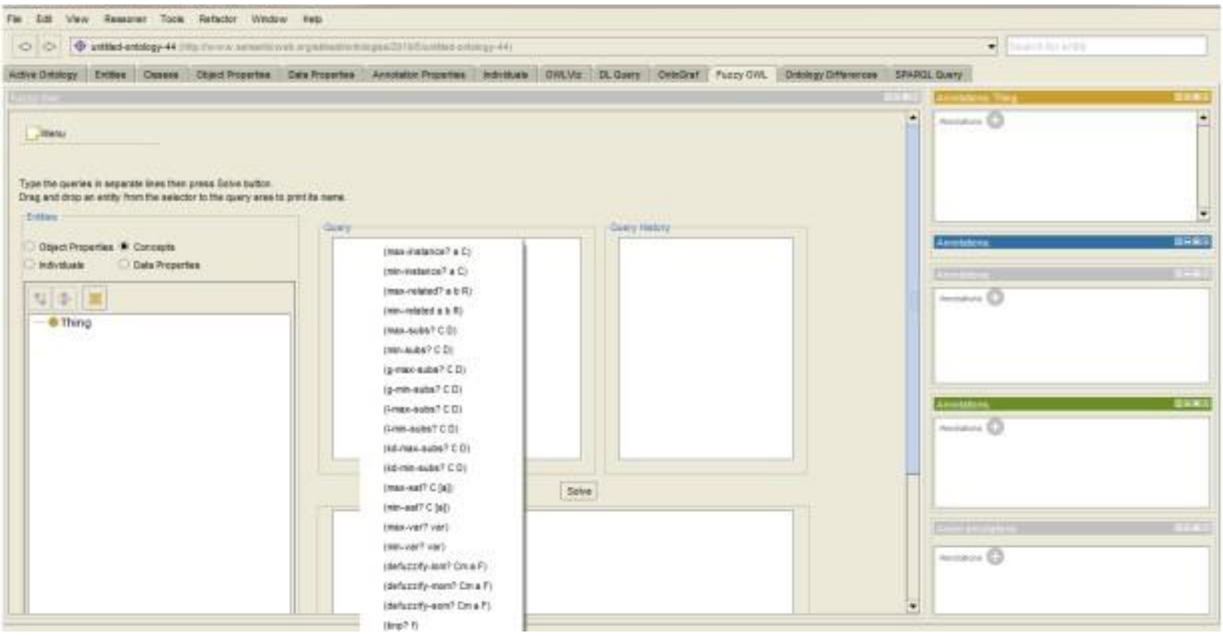
Choose the type and fill with parameters

No Reasoner set. Select a reasoner from the Reasoner menu

Figure 5.4: Fuzzy OWL 2 plugin options menu.

2.3 Fuzzy DL Reasoner

Fuzzy DL Reasoner is a Java based reasoner and deals with fuzzy information, it supports fuzzy logic reasoning and using Java parser generator and MILP-solver. The reasoning algorithm uses a combination of an array algorithm and a MILP optimization problem.



Type the queries in separate lines then press Solve button.
Drag and drop an entity from the selector to the query area to print its name.

Figure 5.5: Queries from fuzzy DL reasoner.

2.4 Eclipse

2.4.1 Definition

Eclipse is an IDE, Integrated Development Environment (EDI Integrated Development Environment in French), that is, software that simplifies programming by offering a number of shortcuts and programming assistance. It is developed by IBM, is free and available for most operating systems.

Figure 5.6: Eclipse logo

2.4.2 Installation Process

The software is already installed on your machines. Just typing Eclipse in a terminal to throw it. It is possible to specify the version of Eclipse that you want to launch via the Eclipse-Kepler, Eclipse-Luna or Eclipse-Mars commands. If you want to install it at home, Eclipse is available on the website http://www.eclipse.org for Linux, BSD, Windows or MacOS. The version to download is the last stable. Normally, the page http://www.eclipse.org/downloads automatically chooses the correct version.

2.5 OWL

For the implementation of our ontology we used OWL ontology language by the protected editor 5.5.0. OWL is a semantic web language designed to represent rich and complex knowledge about things, groups of things, and relationships between things. It is a logic-based calculation language such that knowledge expressed in OWL can be exploited by computer programs. The OWL language includes two versions, OWL1 and OWL2. The OWL 2 language is an ontology language for the Semantic Web in the formally defined sense. OWL 2 was developed by the W3C OWL working group in 2009. OWL1 is the starting point for the development of OWL2 or in other words OWL2 is the development of OWL1. OWL 2 ontologies provide classes, properties, individuals and data values and are stored as semantic web documents.

2.6 Java

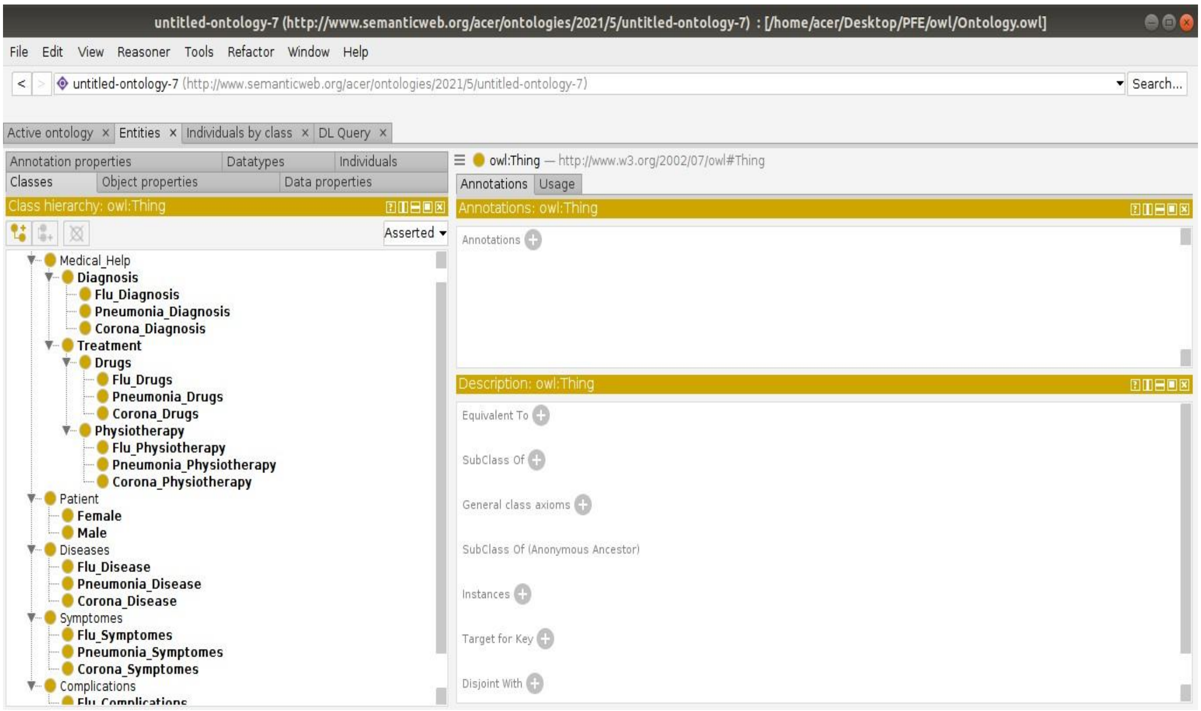
Java is an object-oriented programming language that produces software for multiple platforms. Eclipse and NetBeans are the Java language development environments the most popular.

Figure 5.7: Java logo

3 The implementation Process

Step 1: The Concept's definition

Figure 5.8 represents the hierarchy of concepts used in our ontology.



untitled-ontology-7 (http://www.semanticweb.org/acer/ontologies/2021/5/untitled-ontology-7) : [/home/acer/Desktop/PFE/owl/Ontology.owl]

untitled-ontology-7 (http://www.semanticweb.org/acer/ontologies/2021/5/untitled-ontology-7)

owl:Thing — http://www.w3.org/2002/07/owl#Thing

```
Medical_Help
    Diagnosis
        Flu_Diagnosis
        Pneumonia_Diagnosis
        Corona_Diagnosis
    Treatment
        Drugs
            Flu_Drugs
            Pneumonia_Drugs
            Corona_Drugs
        Physiotherapy
            Flu_Physiotherapy
            Pneumonia_Physiotherapy
            Corona_Physiotherapy
Patient
    Female
    Male
Diseases
    Flu_Disease
    Pneumonia_Disease
    Corona_Disease
Symptomes
    Flu_Symptomes
    Pneumonia_Symptomes
    Corona_Symptomes
Complications
    Flu_Complications
```

Figure 5.8: Class hierarchy.

Step 2: Definition of the relations between the Concepts

Figure 5.9 represents the relations (properties of the object) between the concepts.

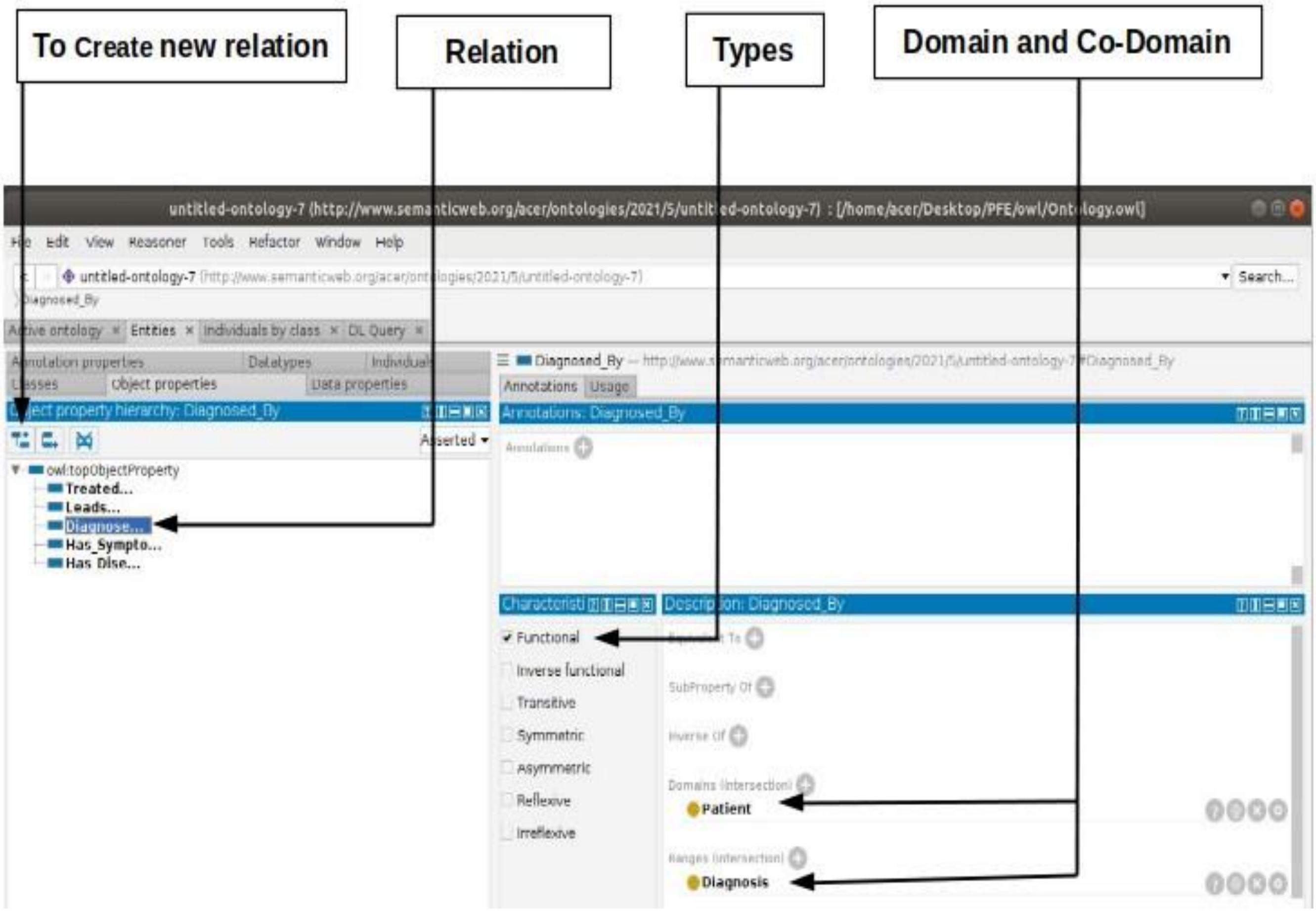

To Create new relation
Relation
Types
Domain and Co-Domain
untitled-ontology-7 (http://www.semanticweb.org/acer/ontologies/2021/5/untitled-ontology-7) : [/home/acer/Desktop/PFE/owl/Ontology.owl]
File Edit View Reasoner Tools Refactor Window Help
Active ontology Entities Individuals by class DL Query
Annotation properties Datatypes Individuals
Classes Object properties Data properties
Object property hierarchy: Diagnosed_By
Asserted
owl:topObjectProperty
Treated...
Leads...
Diagnose...
Has_Sympto...
Has Dise...
Diagnosed_By
Annotations Usage
Annotations: Diagnosed_By
Characteristi
Description: Diagnosed_By
☑ Functional
□ Inverse functional
□ Transitive
□ Symmetric
□ Asymmetric
□ Reflexive
□ Irreflexive
Equivalent To
SubProperty Of
Inverse Of
Domains (intersection)
Patient
Ranges (intersection)
Diagnosis

Figure 5.9: Relations between concepts.

Step 3: The Instantiation of concepts

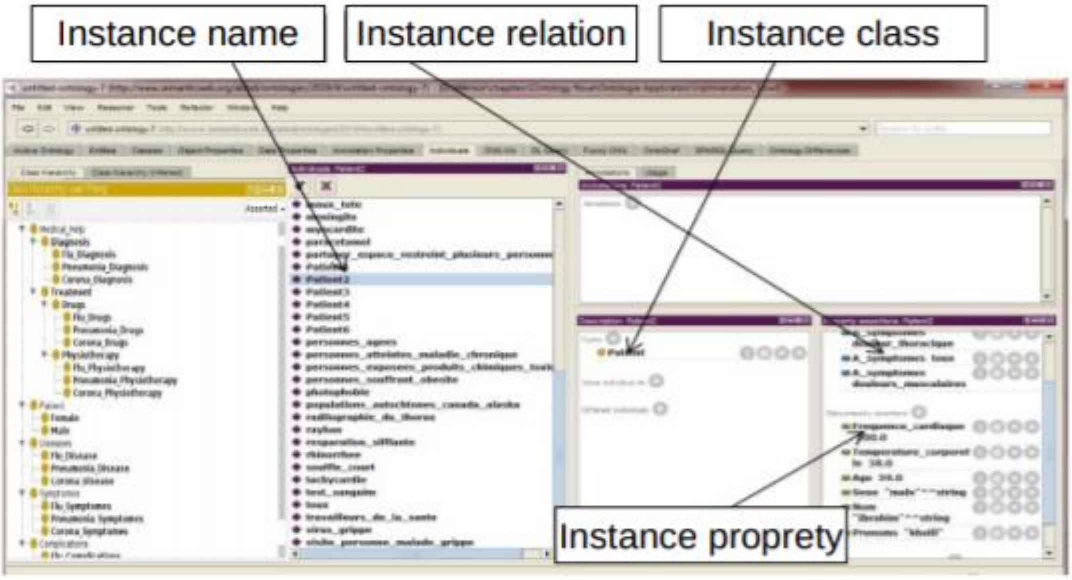
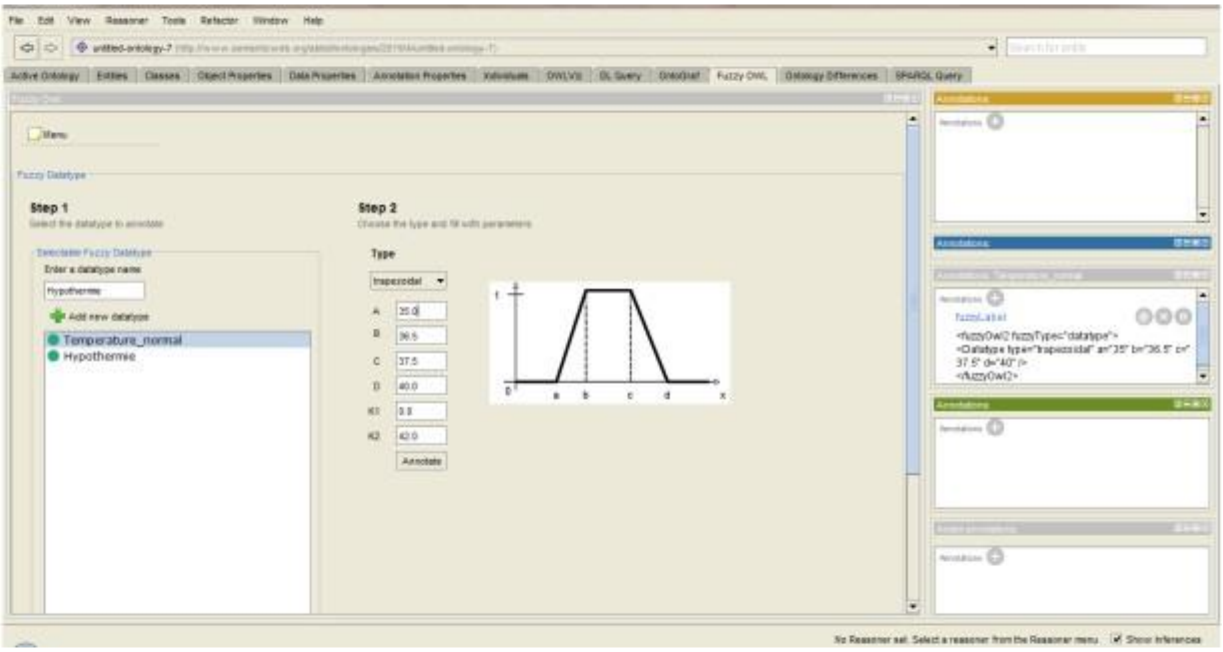


Figure 5.10: The Instances of the Concepts.

Step 4: Data types creation with Fuzzy Owl2 plugin

Figure 5.11 illustrates how the plug-in works by showing how to create a new fuzzy data type. We specify the name of the data type and the type of the membership function. Then the plug-in asks for the necessary parameters depending on the type. An image is displayed to help us remember the meaning of the parameters.


File Edit View Reasoner Tools Refactor Window Help
untitled-ontology-7
Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals OWLViz DL Query OntoGraf Fuzzy OWL Ontology Differences SPARQL Query
Menu
Fuzzy Datatype
Step 1
Step 2
Type
Add new datatype
Annotate
Annotations
fuzzyLabel
0 a b c d x
No Reasoner set. Select a reasoner from the Reasoner menu
Show Inferences

Enter a datatype name: Hypothermie

Temperature_normal
Hypothermie

Type: trapezoidal

A 35.0
B 36.5
C 37.5
D 40.0
K1 0.0
K2 42.0

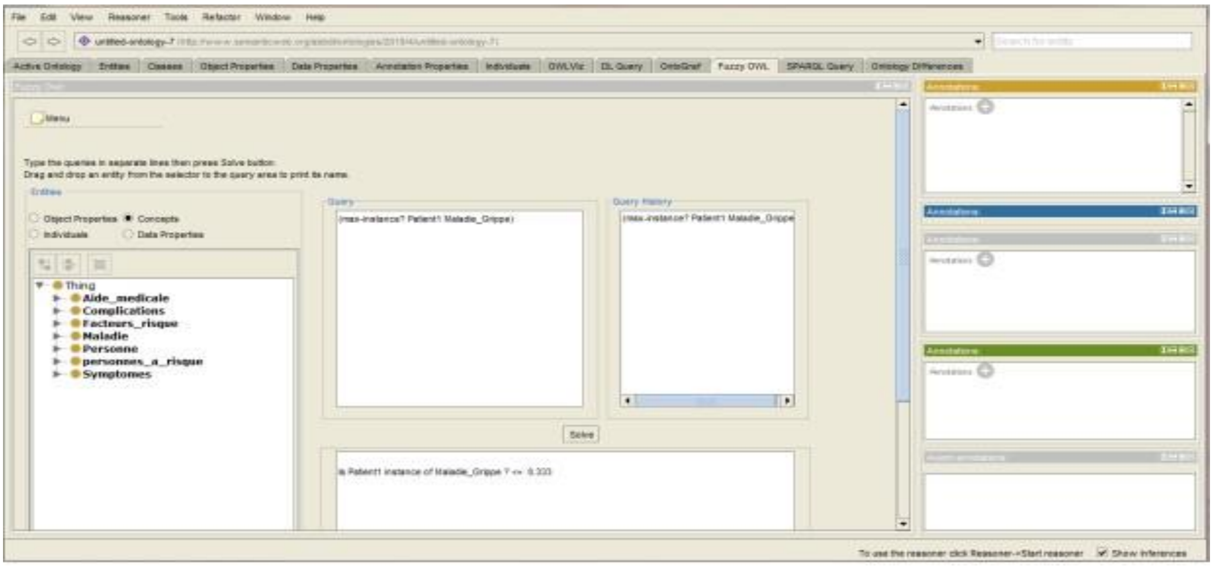
```
<fuzzyOwl2 fuzzyType="datatype">
<Datatype type="trapezoidal" a="35" b="36.5" c="
37.5" d="40" />
</fuzzyOwl2>
```

Figure 5.11: Example of a fuzzy data type Creation with the plug-in.

4 The verification

In this step, we check the validity of our ontology through sending queries using fuzzy DL reasoner.

From the plug-in (Fuzzy OWL 2), we can query the fuzzyDL reasoner which allows submitting requests, provided that these requests are in the form of requests supported by fuzzy DL. This process allows the translation of OWL 2 with annotations representing fuzzy information to the language supported by a reasoner.


File Edit View Reasoner Tools Refactor Window Help
untitled-ontology-7
Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals OWLViz DL Query OntoGraf Fuzzy OWL SPARQL Query Ontology Differences
Menu
Entities
○ Object Properties ● Concepts
○ Individuals ○ Data Properties
Thing
Aide_medicale
Complications
Facteurs_risque
Maladie
Personne
personnes_a_risque
Symptomes
Query
Query History
Solve
Annotations
To use the reasoner click Reasoner->Start reasoner ☑ Show Inferences

Type the queries in separate lines then press Solve button.
Drag and drop an entity from the selector to the query area to print its name.

Query: (max-instance? Patient1 Maladie_Grippe)

Query History: (max-instance? Patient1 Maladie_Grippe

is Patient1 instance of Maladie_Grippe ? <= 0.333

Figure 5.12: Example of Ontology test with fuzzy DL Reasoner Query.

5 Application's GUIs

The Figure 5.13 represents the application's Home:

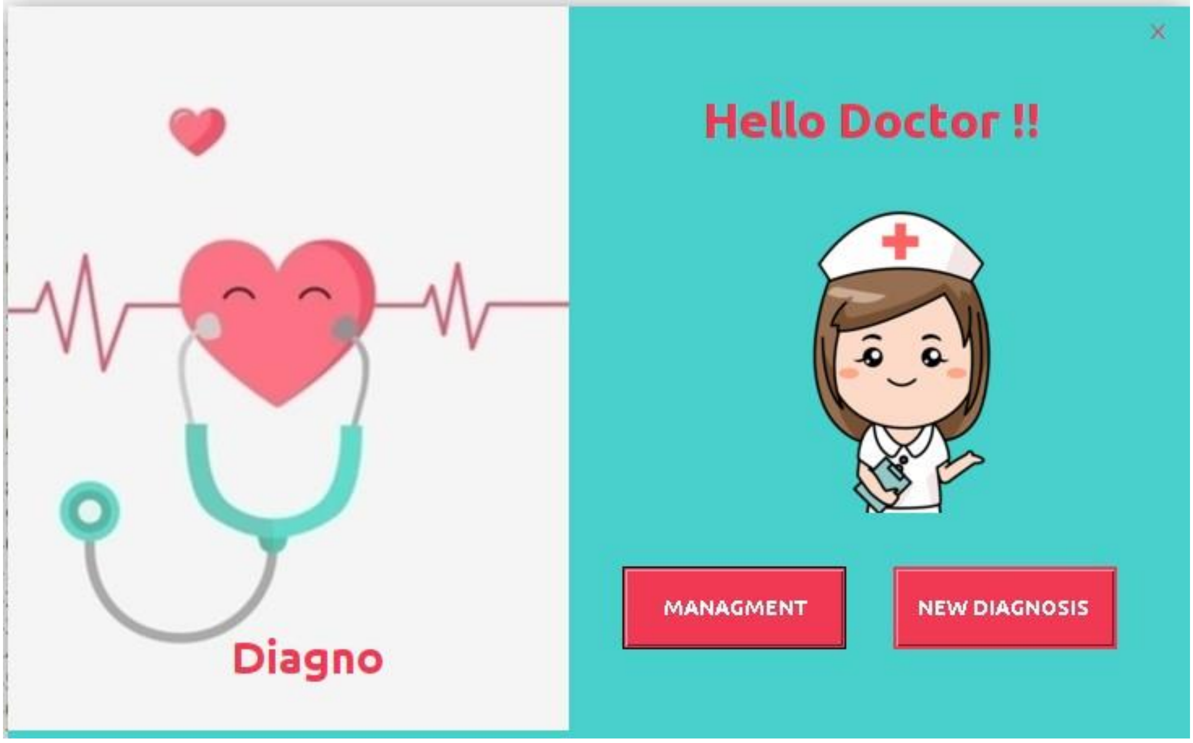
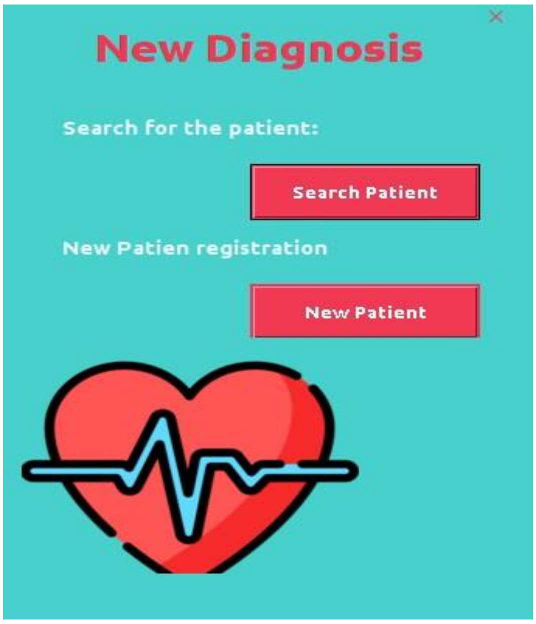


Figure 5.13: Home page interface

The Figure 5.14 represents the interface that offers the doctor search for an existed patient or either to pass for a new patient's registration.



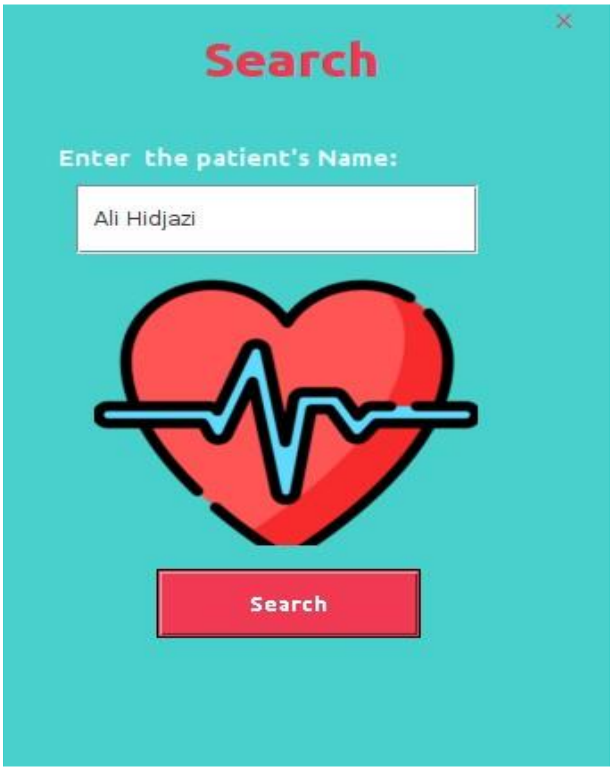
New Diagnosis

Search for the patient:

New Patien registration

Figure 5.14: New diagnosis interface

The Figure 5.15 represents the interface that offers the doctor search for an existed patient to rediagnose him.

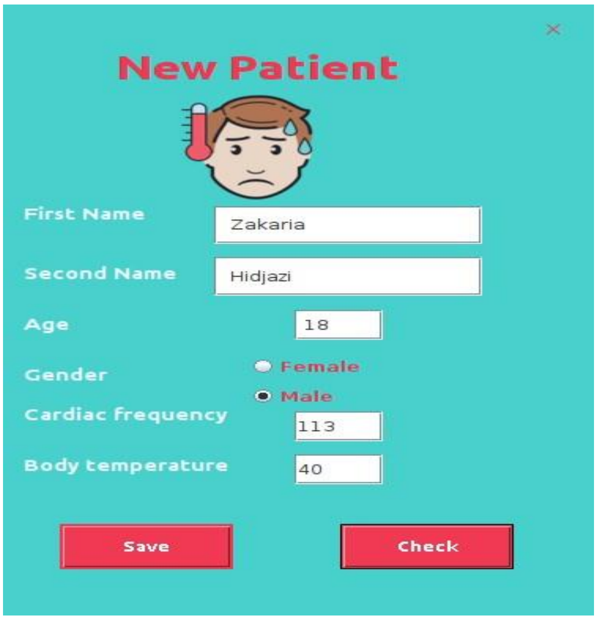


Search

Enter the patient's Name: Ali Hidjazi

Figure 5.15: Search for an existed patient Interface

The Figure 5.16 represents a form for a new patient’s registration.



First Name: Zakaria

Second Name: Hidjazi

Age: 18

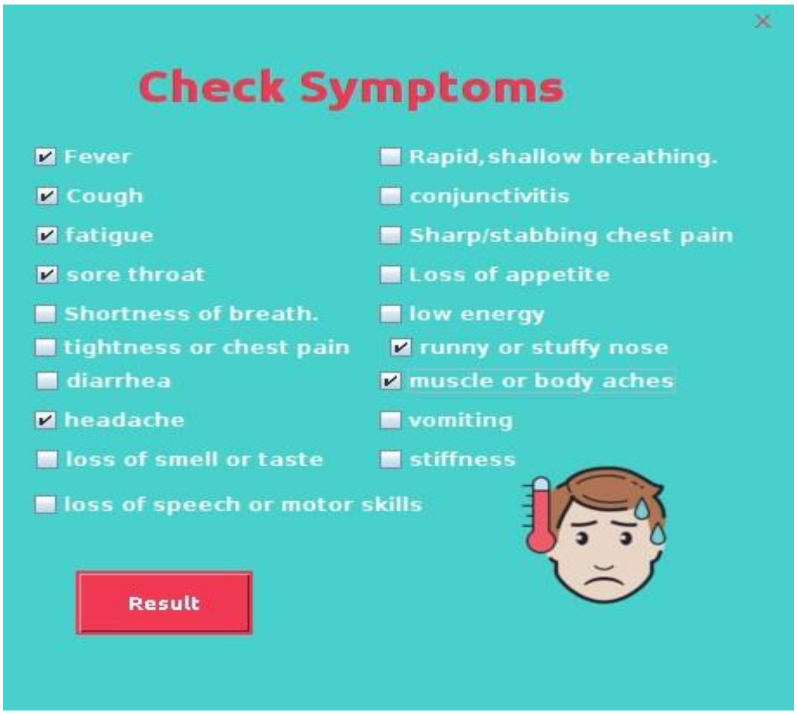
Gender: ○ Female ● Male

Cardiac frequency: 113

Body temperature: 40

Figure 5.16: New patient interface

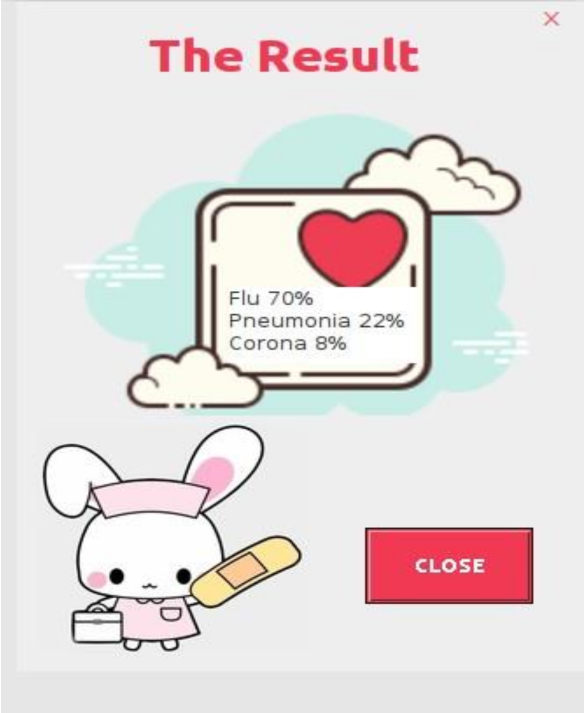
The Figure 5.17 represents the Diagnosis interface where the doctor can check all the symptoms that appeared on the patient and then click result button ”Result” to discover the Diagnosis ‘s result.



- ☑ Fever
- ☑ Cough
- ☑ fatigue
- ☑ sore throat
- □ Shortness of breath.
- □ tightness or chest pain
- □ diarrhea
- ☑ headache
- □ loss of smell or taste
- □ loss of speech or motor skills
- □ Rapid, shallow breathing.
- □ conjunctivitis
- □ Sharp/stabbing chest pain
- □ Loss of appetite
- □ low energy
- ☑ runny or stuffy nose
- ☑ muscle or body aches
- □ vomiting
- □ stiffness

Figure 5.17: Check symptoms interface

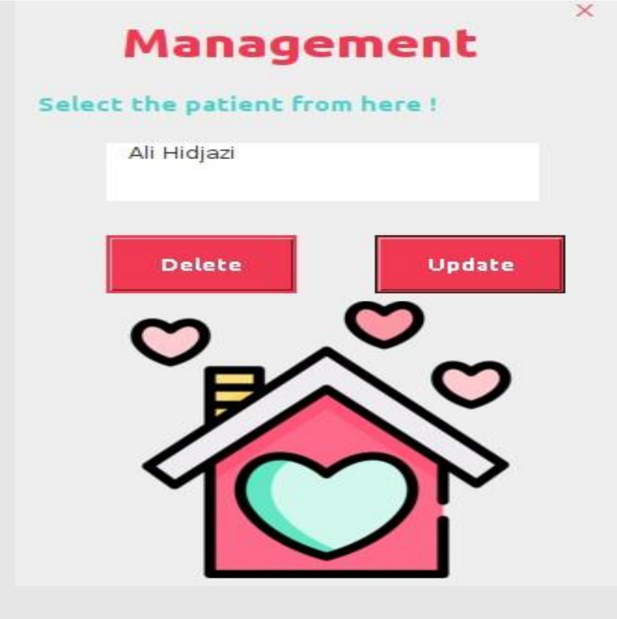
The Figure 5.18 represents the interface that shows the result of the diagnosis



Flu 70%
Pneumonia 22%
Corona 8%

Figure 5.18: Diagnosis result Interface

The Figure 5.19 represents the Management interface where the doctor can delete a patient or update his information.



Ali Hidjazi

Figure 5.19: Management Interface

6 Conclusion

In this chapter, we introduced the details in which we created our ontology in addition to the used tools in this work as well. After that we introduced some GUIs that belong to our application explaining the role of each one.

General Conclusion

In this work, we have implemented a medical diagnostic help system using a fuzzy ontology, we presented in details the notions related to: ontologies, fuzzy logic as well as fuzzy ontologies.

We have also mentioned the importance of fuzzy ontologies in solving the problems of classical ontology by allowing the reasoning and representation of imprecise and uncertain knowledge.

In order to build a fuzzy ontology that makes our system capable to help the doctor diagnose correctly, we presented the creation method of this ontology, we also collected a set of information related to the medical diagnosis, such as diseases, symptoms ... etc, where the purpose is to represent this knowledge.

So to build our ontology we have chosen the "METHONTOLOGY" method, because it is a famous method used by many ontology developers and the more important that it is the best suited to our ontology’s construction.

By using the fuzzy logic we have manipulated fuzzy concepts and relations, It is by identifying the points of uncertainty and imprecision which present in these concepts and these ontology relations to generate fuzzy concepts and relations allowing to represent fuzzy information and fuzzy data for the disease.

By using the "protégé" editor we have in result a fuzzy ontology. Then we confirmed the validity of our fuzzy ontology using the fuzzy DL reasoner which belongs to the "plug-in Fuzzy OWL2", at the end we programmed a simple system which manages this ontology and make it easy to use.

Bibliography

[1] (Site 1), Retrieved: 17/06/2021 18:01, From: https://philosophyterms.com/ontology/

[2] (Site 2), Retrieved: 17/06/2021 18:04, From : https://en.wikipedia.org/wiki/Ontology_(information_science)#History

[3] Espinasse Bernard, « Introduction aux Ontologies », University of Aix-Marseille 2010.

[4] Declan O'Sullivan, «Brief Introduction to Ontology» , University of Dublin Trinity College , 2012.

[5] Guarino Nicola, Oberle Daniel, Staab Steffen, «What Is an Ontology ? », In book: Handbook on Ontologies , pp.1-17 , may 2009.

[6] Reyes-Peña Cecilia, Tovar-Vidal Mireya, «Ontology: Components and Evaluation, a Review». Benemérita Universidad Autónoma de Puebla, Faculty of Computer Science, Puebla, Mexico, pp. 257–265, 2018.

[7] (site 3), Retrieved : 27/02/2021 ,17:58, From: https://en.wikipedia.org/wiki/Fuzzy_logic

[8] (site 4), Retrieved : 27/02/2021 ,17:58, From : https://plato.stanford.edu/entries/logic-fuzzy/

[9] (Site 5), Retrieved : 27/03/2021 ,21:58, From : https://www.upgrad.com/blog/fuzzy-login-in-artificial-intelligence/#Fuzzy_Logic_Architecture

[10] Dernoncourt F, «Fuzzy Logic: Between Human Reasoning and Artificial Intelligence» Report; Ecole Normale Supperieure: Paris, France, 2011.

[11] Chevrie François, Guely François, « Fuzzy Logic:cahier téchnique ».GROUPE SCHNEIDER,1998, N°191.

[12] Agrebi Hanene, Bahri Afef, Bouaziz Rafik, «Fuzzy Ontologies Model for Semantic», The Second International Conférence on Information and Knowledge Management, eKNow 2010, Février 2010, st. Maorten, Netherlands Antilles .

[13] Fullér Robert, «What is Fuzzy Logic and Fuzzy Ontology? », KnowMobile National Workshop, October 30, 2008, Helsinki.

[14] Riahi Samani Zahra, Shamsfard Mehrnoush, «The State of the Art in Developing Fuzzy Ontologies: A survey» book, Faculty of Computer Science and Engineering, Shahid Beheshti University, 2018.

[20] Ladjenef Mohamed, « système à base d'ontologie pour l'aide à la maintenance des machines », Thesis submitted for obtaining the degree of Academic Master in Computer Science, University of Mohamed Boudiaf m'sila, 2017/2018.

[18] (Site 6), Retrieved : 16/06/2021, 01:34, From: https://www.passeportsante.net/fr/Maux/Problemes/Fiche.aspx?doc=grippe_pm

[17] (Site 7), Retrieved : 16/06/2021, 01:22, From : https://www.passeportsante.net/fr/Maux/Problemes/Fiche.aspx?doc=pneumonie_pm

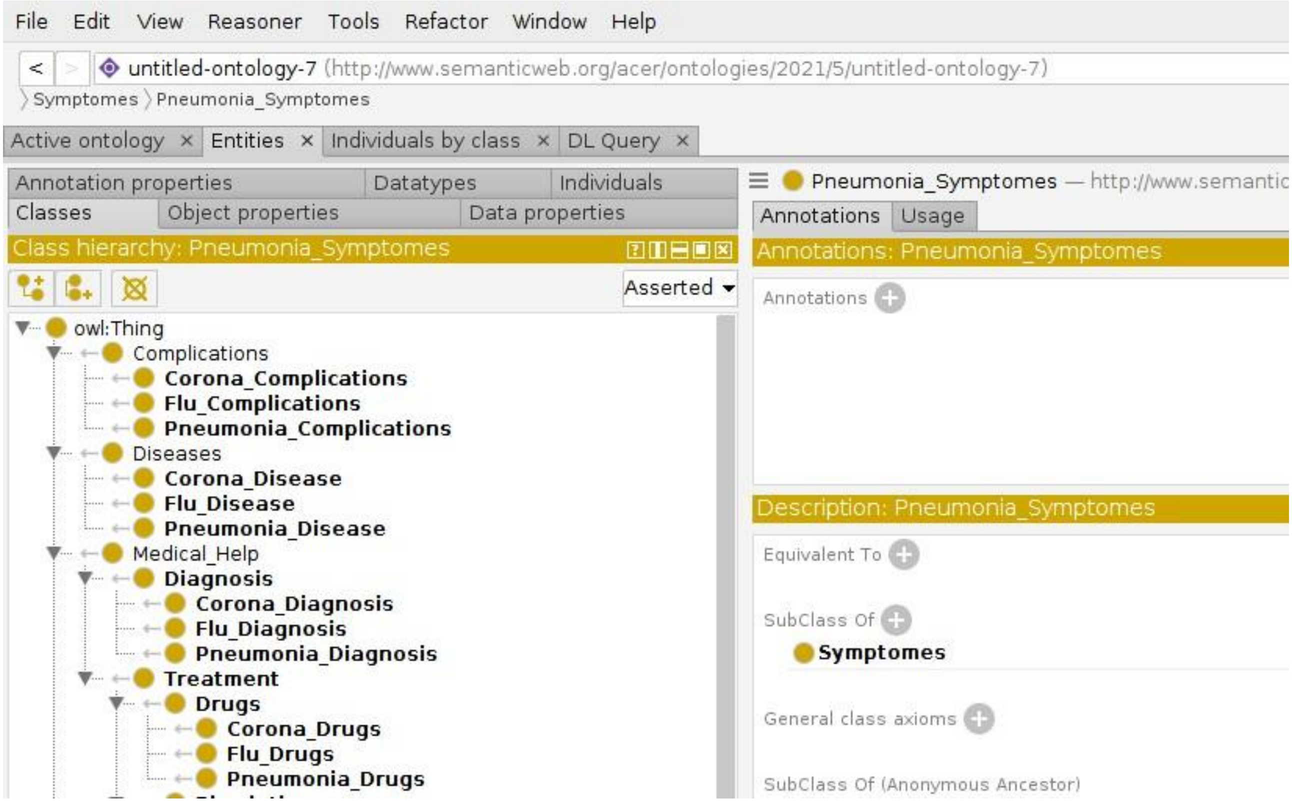
[19] (Site 8), Retrieved : 16/06/2021, 01:42, From: https://www.futura-sciences.com/sante/definitions/coronavirus-covid-19-18585/

The Application Annex

1 Source Code

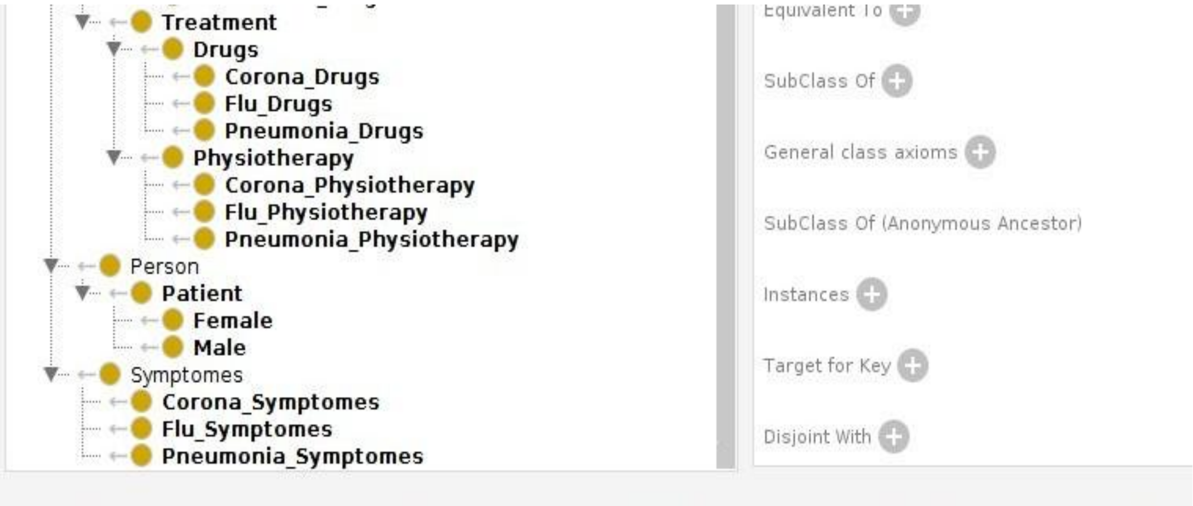
1.1 The Hierarchy of the created ontology

The next figures show the classes and the sub-classes of our ontology


File
Edit
View
Reasoner
Tools
Refactor
Window
Help
untitled-ontology-7 (http://www.semanticweb.org/acer/ontologies/2021/5/untitled-ontology-7)
Symptomes 〉 Pneumonia_Symptomes
Active ontology ×
Entities ×
Individuals by class ×
DL Query ×
Annotation properties
Datatypes
Individuals
Classes
Object properties
Data properties
Class hierarchy: Pneumonia_Symptomes
Asserted
Pneumonia_Symptomes — http://www.semantic
Annotations
Usage
Annotations: Pneumonia_Symptomes
Annotations
Description: Pneumonia_Symptomes
Equivalent To
SubClass Of
General class axioms
SubClass Of (Anonymous Ancestor)

- owl:Thing
  - Complications
    - Corona_Complications
    - Flu_Complications
    - Pneumonia_Complications
  - Diseases
    - Corona_Disease
    - Flu_Disease
    - Pneumonia_Disease
  - Medical_Help
    - Diagnosis
      - Corona_Diagnosis
      - Flu_Diagnosis
      - Pneumonia_Diagnosis
    - Treatment
      - Drugs
        - Corona_Drugs
        - Flu_Drugs
        - Pneumonia_Drugs

SubClass Of: Symptomes



- Treatment
  - Drugs
    - Corona_Drugs
    - Flu_Drugs
    - Pneumonia_Drugs
  - Physiotherapy
    - Corona_Physiotherapy
    - Flu_Physiotherapy
    - Pneumonia_Physiotherapy
- Person
  - Patient
    - Female
    - Male
- Symptomes
  - Corona_Symptomes
  - Flu_Symptomes
  - Pneumonia_Symptomes

1.2 The Code of the created ontology

```
<?xml version="1.0"?>
<rdf:RDF xmlns="
ontology-7#"
    xml:base="http://www.semanticweb.org/acer/ontologies/2021/5/untitled-ontology-7"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
<owl:Ontology rdf:about="
untitled-ontology-7"/>
<!--
   ///////////////////////////////////////////////////////////////////////////////////////
   //
   // Object Properties
   //
   ///////////////////////////////////////////////////////////////////////////////////////
    -->
<!--
#Diagnosed_By -->
<owl:ObjectProperty rdf:about="
/untitled-ontology-7#Diagnosed_By">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
<rdfs:domain rdf:resource="
/5/untitled-ontology-7#Patient"/>
<rdfs:range rdf:resource="
5/untitled-ontology-7#Diagnosis"/>
</owl:ObjectProperty>
<!--
#Has_Disease -->
<owl:ObjectProperty rdf:about="
5/untitled-ontology-7#Has_Disease">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
<rdfs:domain rdf:resource="
/5/untitled-ontology-7#Patient"/>
<rdfs:range rdf:resource="
5/untitled-ontology-7#Diseases"/>
</owl:ObjectProperty>
<!--
#Has_Symptomes -->
<owl:ObjectProperty rdf:about="
5/untitled-ontology-7#Has_Symptomes">
<rdf:type rdf:resource="
#FunctionalProperty"/>
```

```
<rdfs:domain rdf:resource="
2021/5/untitled-ontology-7#Diseases"/>
<rdfs:range rdf:resource="
2021/5/untitled-ontology-7#Symptomes"/>
</owl:ObjectProperty>
<!--
#Leads_To -->
<owl:ObjectProperty rdf:about="
2021/5/untitled-ontology-7#Leads_To">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
<rdfs:domain rdf:resource="
2021/5/untitled-ontology-7#Diseases"/>
<rdfs:range rdf:resource="
2021/5/untitled-ontology-7#Complications"/>
</owl:ObjectProperty>
<!--
untitled-ontology-7#Treated_By -->
<owl:ObjectProperty rdf:about="
2021/5/untitled-ontology-7#Treated_By">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
<rdfs:domain rdf:resource="
2021/5/untitled-ontology-7#Patient"/>
<rdfs:range rdf:resource="
2021/5/untitled-ontology-7#Treatment"/>
</owl:ObjectProperty>
<!-- ///////////////////////////////////////////////////////////////////////////////////////
   //
   // Classes
   //
 ///////////////////////////////////////////////////////////////////////////////////////
    →
<!--
#Complications -->
<owl:Class rdf:about="
untitled-ontology-7#Complications"/>
<!--
#Corona_Complications -->
<owl:Class rdf:about="
5/untitled-ontology-7#Corona_Complications">
<rdfs:subClassOf rdf:resource="
2021/5/untitled-ontology-7#Complications"/>
</owl:Class>
<!--
#Corona_Diagnosis -->
<owl:Class rdf:about="
2021/5/untitled-ontology-7#Corona_Diagnosis">
```

```
<rdfs:subClassOf rdf:resource="
ontologies/2021/5/untitled-ontology-7#Diagnosis"/>
</owl:Class>
<!--
#Corona_Disease -->
<owl:Class rdf:about="
untitled-ontology-7#Corona_Disease">
<rdfs:subClassOf rdf:resource="
2021/5/untitled-ontology-7#Diseases"/>
</owl:Class>
<!--
#Corona_Drugs -->
<owl:Class rdf:about="
untitled-ontology-7#Corona_Drugs">
<rdfs:subClassOf rdf:resource="
2021/5/untitled-ontology-7#Drugs"/>
</owl:Class>
<!--
#Corona_Physiotherapy -->
<owl:Class rdf:about="
untitled-ontology-7#Corona_Physiotherapy">
<rdfs:subClassOf rdf:resource="
2021/5/untitled-ontology-7#Physiotherapy"/>
</owl:Class>
<!--
#Corona_Symptomes -->
<owl:Class rdf:about="
untitled-ontology-7#Corona_Symptomes">
<rdfs:subClassOf rdf:resource="
ontologies/2021/5/untitled-ontology-7#Symptomes"/>
</owl:Class>
<!--
#Diagnosis -->
<owl:Class rdf:about="
untitled-ontology-7#Diagnosis">
<rdfs:subClassOf rdf:resource="
2021/5/untitled-ontology-7#Medical_Help"/>
</owl:Class>
<!--
#Diseases -->
<owl:Class rdf:about="
untitled-ontology-7#Disease
<!--
#Drugs -->
<owl:Class rdf:about="
untitled-ontology-7#Drugs">
```

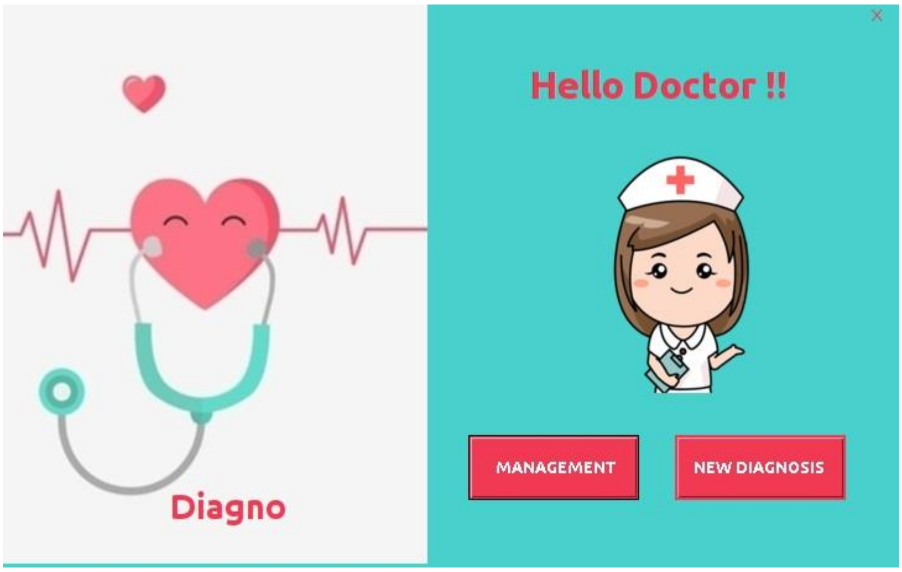
```
<rdfs:subClassOf rdf:resource="
ontologies/2021/5/untitled-ontology-7#Treatment"/>
</owl:Class>
<!--
#Female -->
<owl:Class rdf:about="
untitled-ontology-7#Female">
<rdfs:subClassOf rdf:resource="
ontologies/2021/5/untitled-ontology-7#Patient"/>
</owl:Class>
<!--
#Flu_Complications -->
<owl:Class rdf:about="
untitled-ontology-7#Flu_Complications">
<rdfs:subClassOf rdf:resource="
ontologies/2021/5/untitled-ontology-7#Complications"/>
</owl:Class>
<!--
untitled-ontology-7#Flu_Diagnosis -->
<owl:Class rdf:about="
untitled-ontology-7#Flu_Diagnosis">
<rdfs:subClassOf rdf:resource="
ontologies/2021/5/untitled-ontology-7#Diagnosis"/>
</owl:Class>
<!--
#Flu_Disease -->
<owl:Class rdf:about="
untitled-ontology-7#Flu_Disease">
<rdfs:subClassOf rdf:resource="
2021/5/untitled-ontology-7#Diseases"/>
</owl:Class>
<!--
#Flu_Drugs -->
<owl:Class rdf:about="
untitled-ontology-7#Flu_Drugs">
<rdfs:subClassOf rdf:resource="
2021/5/untitled-ontology-7#Drugs"/>
</owl:Class>
<!--
#Flu_Physiotherapy -->
<owl:Class rdf:about="
untitled-ontology-7#Flu_Physiotherapy">
<rdfs:subClassOf rdf:resource="
2021/5/untitled-ontology-7#Physiotherapy"/>
</owl:Class>
<!--
```

```
#Flu_Symptomes -->
<owl:Class rdf:about="
untitled-ontology-7#Flu_Symptomes">
<rdfs:subClassOf rdf:resource="
2021/5/untitled-ontology-7#Symptomes"/>
</owl:Class>
<!-- http://www.semanticweb.org/acer/ontologies/2021/5/untitled-ontology-7#Male -->
<owl:Class rdf:about="
untitled-ontology-7#Male">
<rdfs:subClassOf rdf:resource="
2021/5/untitled-ontology-7#Patient"/>
</owl:Class>
<!--
#Medical_Help -->
<owl:Class rdf:about="
untitled-ontology-7#Medical_Help"/>
<!--
#Patient -->
<owl:Class rdf:about="
untitled-ontology-7#Patient">
<rdfs:subClassOf rdf:resource="
2021/5/untitled-ontology-7#Person"/>
</owl:Class>
<owl:Class rdf:about="
untitled-ontology-7#Person"/>
<!--
#Physiotherapy -->
<owl:Class rdf:about="
untitled-ontology-7#Physiotherapy">
<rdfs:subClassOf rdf:resource="
ontologies/2021/5/untitled-ontology-7#Treatment"/>
</owl:Class>
<!--
#Pneumonia_Complications -->
<owl:Class rdf:about="
untitled-ontology-7#Pneumonia_Complications">
<rdfs:subClassOf rdf:resource="
2021/5/untitled-ontology-7#Complications"/>
</owl:Class>
<!--
#Pneumonia_Diagnosis -->
<owl:Class rdf:about="
untitled-ontology-7#Pneumonia_Diagnosis">
<rdfs:subClassOf rdf:resource="
2021/5/untitled-ontology-7#Diagnosis"/>
</owl:Class>
```

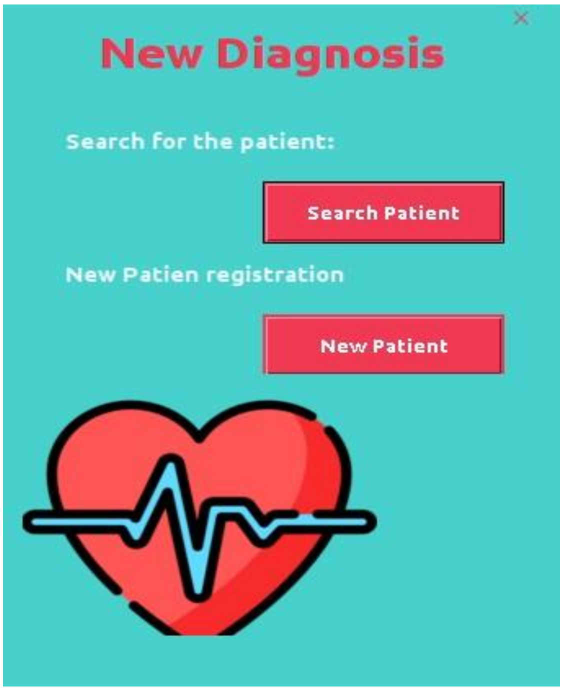
```
<!--
#Pneumonia_Disease -->
<owl:Class rdf:about="
untitled-ontology-7#Pneumonia_Disease">
<rdfs:subClassOf rdf:resource="
ontologies/2021/5/untitled-ontology-7#Diseases"/>
</owl:Class>
<!--
#Pneumonia_Drugs -->
<owl:Class rdf:about="
untitled-ontology-7#Pneumonia_Drugs">
<rdfs:subClassOf rdf:resource="
ontologies/2021/5/untitled-ontology-7#Drugs"/>
</owl:Class>
<!--
#Pneumonia_Physiotherapy -->
<owl:Class rdf:about="
5/untitled-ontology-7#Pneumonia_Physiotherapy">
<rdfs:subClassOf rdf:resource="
2021/5/untitled-ontology-7#Physiotherapy"/>
</owl:Class>
<!--
#Pneumonia_Symptomes -->
<owl:Class rdf:about="
untitled-ontology-7#Pneumonia_Symptomes">
<rdfs:subClassOf rdf:resource="
ontologies/2021/5/untitled-ontology-7#Symptomes"/>
</owl:Class>
<!--
#Symptomes -->
<owl:Class rdf:about="
untitled-ontology-7#Symptomes"/>
<!--
#Treatment -->
<owl:Class rdf:about="
2021/5/untitled-ontology-7#Treatment">
<rdfs:subClassOf rdf:resource="
2021/5/untitled-ontology-7#Medical_Help"/>
</owl:Class>
</rdf:RDF>
<!-- Generated by the OWL API (version 4.5.9.2019-02-01T07:24:44Z)
https://github.com/owlcs/owlapi -->
```

2 The Application's Running Method

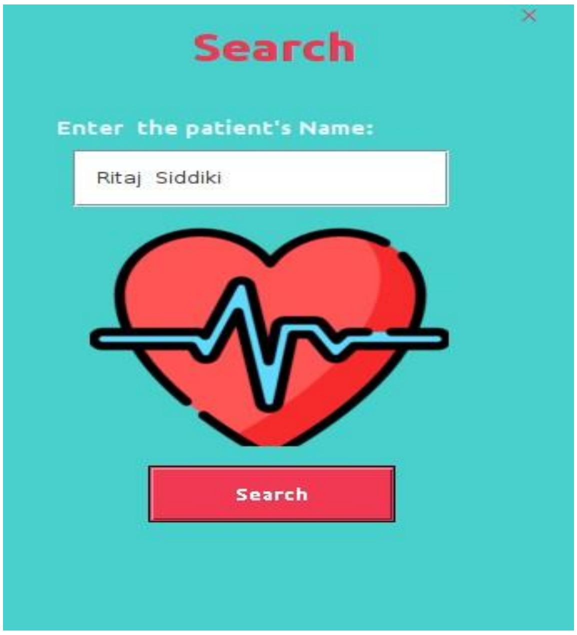
The next figure shows the home of the application, where the doctor can pass to patient diagnosis or either to manage the system:



The next figure shows the user interface that allows to the doctor the search for rediagnose him or the registration for a new patient:



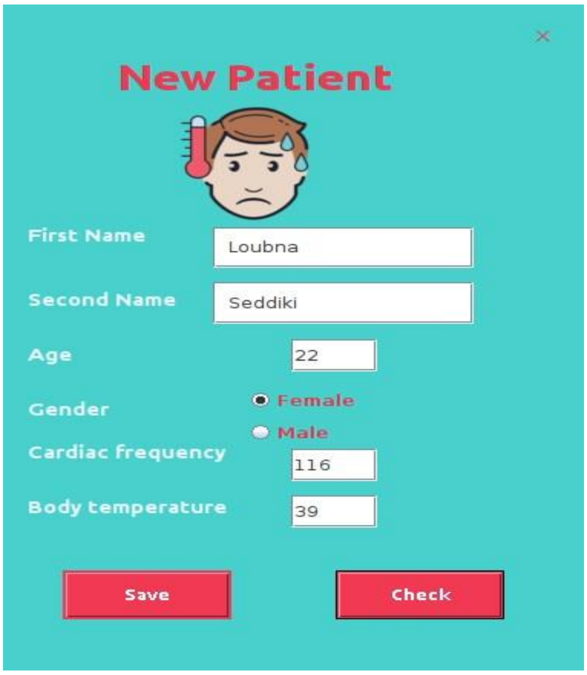
The next figure represents the user interface that allows the doctor to search for a patient:



Search

Enter the patient's Name: Ritaj Siddiki

The next figure represents a new patient’s registration form:



New Patient

First Name: Loubna

Second Name: Seddiki

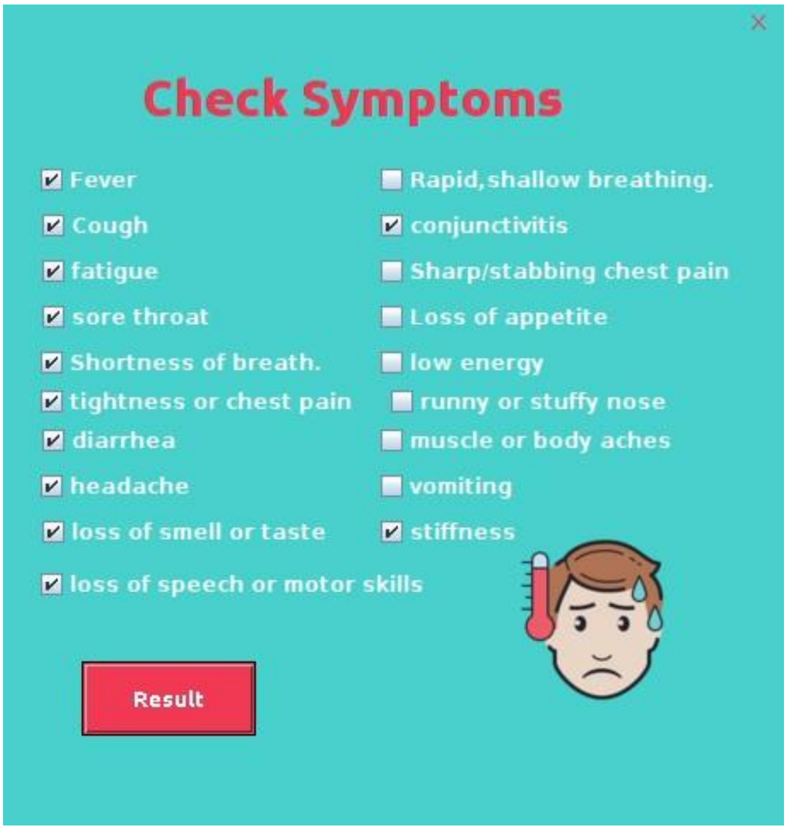
Age: 22

Gender: ● Female ○ Male

Cardiac frequency: 116

Body temperature: 39

The next figure represents the diagnosis platform, where the doctor checks the symptoms of the patient and then discovers the disease by clicking on the button “result”:

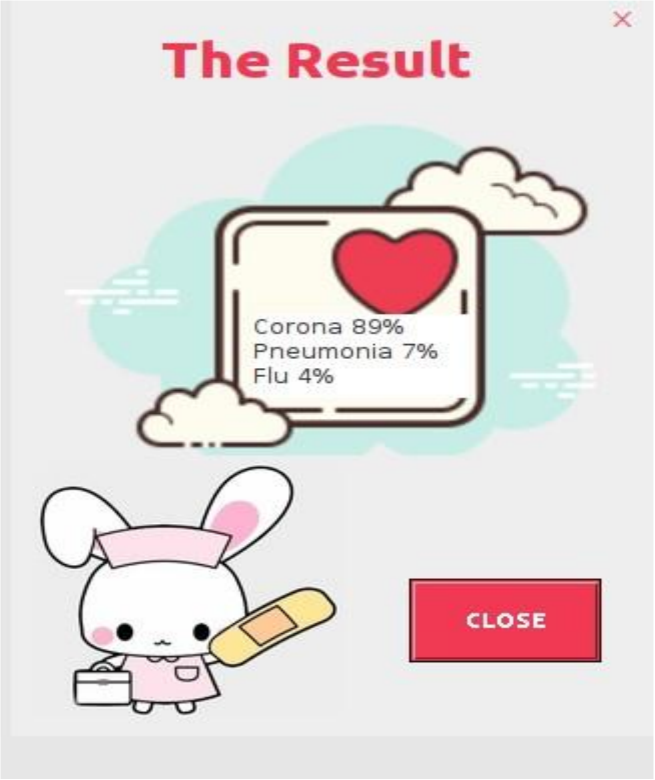


Check Symptoms

☑ Fever
☑ Cough
☑ fatigue
☑ sore throat
☑ Shortness of breath.
☑ tightness or chest pain
☑ diarrhea
☑ headache
☑ loss of smell or taste
☑ loss of speech or motor skills

□ Rapid,shallow breathing.
☑ conjunctivitis
□ Sharp/stabbing chest pain
□ Loss of appetite
□ low energy
□ runny or stuffy nose
□ muscle or body aches
□ vomiting
☑ stiffness

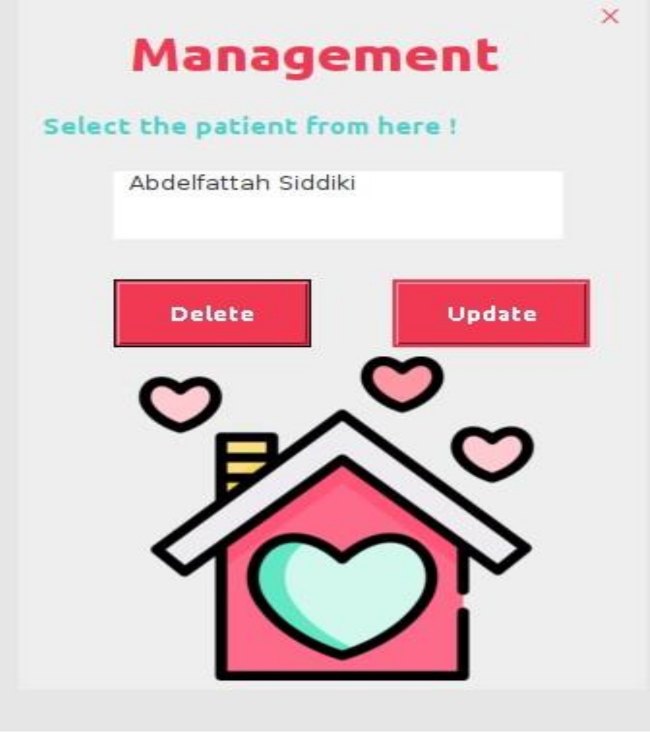
The next figure represents the interface that shows the result of the diagnosis:



The Result

Corona 89%
Pneumonia 7%
Flu 4%

The next figure shows the management interface where the doctor can update the patient‘s information or delete it:



Abdelfattah Siddiki