Algerian Republic Democratic and Popular
Ministry of Higher Education and Scientific Research
University of 8 May 1945-Guelma-
Faculty of Mathematics,Computer Science and Science of Matter
Department of Computer Science

Master Thesis

Specialty : Computer Science

Option :

Science and technology of information and communication

Theme

A Solution based on semantic IoT and machine learning to improve the lives of people with disabilities

---

Encadré Par :
Dr Halimi Khaled

Présenté par :
*B*adji Ahmed Amin

Membres du jury :
Dr Boughareb Djalila
Dr Brahimi Said

June 2022

ACKNOWLEDGEMENTS

First and foremost, I would like to thank God for giving me the strength, knowledge, ability, and opportunity to undertake this research study and persevere and complete it satisfactorily.

I owe a lot to my little family especially my parents, without them I would never reach this level, may god bless them.

I would like to sincerely thank my thesis supervisor, **Dr. Khaled HALIMI**, for my graduate study's outstanding orientation. Thank you for teaching and advising me for the most years of my studies.

I owe a lot to my co-supervisor **Mr. Halim HADJAJI** and **Mr. Adel BOUGHIDA** who have helped, encouraged and guided me a lot to accomplish this work. I thank them for their numerous remarks and appreciations of extreme relevance, and I would like to express my deep gratitude to them on this occasion.

I wish to express my deep gratitude to **Dr. Djalila BOUGHAREB** and to **Dr. Said BRAHIMI** for the honour they have bestowed on me by accepting the responsibility of examining this work and participating in the defence jury.

I owe a lot to **Dr. Zineddine KOUAHLA**, who gave me the benefit of his immense experience and encouraged me to progress in my work and changed my mindset during my years of study in this university. I thank him infinitely for his help, and I express my deepest gratitude.

I would like to express my deep gratitude to **Mr. Charaf NEBILI** for his invaluable help and support during all my years of study. I would also like to thank all the teachers and staff of the Computer Science department of the university of 8 May 1945 Guelma.

My sincere thanks to my friends who was there for me always. **Hamed, Saber, Oussama, Badri, and Jalil**.

ABSTRACT

The technology is in a remarkable and continuous development field, it took a part in all other fields such as medicine which makes it indispensable because of the facilities it brings the doctors and also the patients especially to persons with disabilities. Because of its inter-connectivity, the Internet of Things (IoT) technology has received a lot of attention in recent years. The need for semantics comes from the different sources and sensors we depend on for information collection. In this project we also use Semantic Web technologies (Ontologies, SWRL, SPARQL) to make a combination with database data to ensure better interoperability and thus make the Internet of Things semantic. The large size of the IoT data and the considerable number of SWRL rules requires optimization techniques, so we used machine learning to optimize this process and reduce the number of rules. From this combination, a system based on an enriched ontological model containing knowledge about the target person (such as: vital signs, type of disability, their goals, the obstacles that can be found, etc.) is realized in order to provide a better life to handicaps as well as to give them a little more autonomy in their movements.

Keywords: Internet of Things, Semantic Web, Ontology, SWRL, SPARQL, Interoperability, Semantic Web of Things, Machine Learning, Handicaps

Résumé

La technologie est dans un développement remarquable et continu, elle est entré dans tous les domaines telle que la médecine ce qui la rend indispensable en raison des facilités qu'elle apporte les médecins et aussi les patients notamment aux patients handicapées.

A cause de son interconnectivité, la technologie de l'Internet des objets (IoT) a reçu beaucoup d'attention ces dernières années, Le besoin de sémantique vient des différentes sources et capteurs dont nous dépendons pour la collecte d'informations

Dans ce projet nous utilisons également les technologies du Web sémantique (Ontologies, SWRL, SPARQL) pour faire une combinaison avec les données des bases de données afin de garantir une meilleure interopérabilité et ainsi rendre l'Internet des objets sémantique. La taille volumineuse des données de l'IoT et le nombre consederable des SWRL rules nécessite des techniques d'optimisation, Alors Nous avons utilisé le machine learning pour l'optimisation de ce processus et de réduire le nombre des rules. A partir de cette combinaison, un système basé sur un modèle ontologique enrichi contenant des connaissances sur la personne cible (telles que : signes vitaux, type de type de handicap, leur objectifs, les obstacles que l'on peut trouver, etc.) est réalisé afin d'offrir une meilleure vie aux personnes handicapées ainsi que de leur donner un peu plus d'autonomie dans leurs mouvements.

Mots-clé: Internet des Objets, Web Semantic, Ontologies, SWRL, SPARQL, Interoperability, Web Semantic des Objets, Machine Learning, Handicaps

.

Contents

List of Tables

List of Figures

General Introduction

The Internet of Things is a new technology that has come to light in recent years. The fundamental characteristic of the Internet of Things is that it turns objects into intelligent devices. It can generate and send data over the network and connect to each other without human intervention. All this contributes to improving life and encourage many fields to start integrating these technologies. The demand for the Internet of Things is leading to an increase in the number of connected objects, according to estimates by the research forecasts indicate that by 2025, more than 75 billion objects will be connected to the internet, which will cause a considerable increasing of the amount of data. Moreover, This expansion has created major technical challenges such as the lack of interoperability, since Internet of Things (IoT) data comes from a variety of sources, applications that use this data suffers. The semantic web is an adequate solution to this problem. But it has its limitations as well, the huge amount of data collected from IoT devices needs to be processed using semantic reasoning engines, which also uses some rules to infer the new knowledge, it called SWRL rules. All of this process needs to be optimized to provide the best results in the smallest period of time. In this project, we aim to optimize the process of semantic reasoning with new architecture that contains a machine learning algorithm, The proposed solution is applied in the field of the health of people with disabilities in order to eliminate the difficulties they encounter in their lives.

Chapter 1

Semantic Web and IoT

1.1 introduction

Nowadays, technology has become one of the most important elements that facilitate human's daily life. As well as the persons with disabilities. In this context, we will talk about them, their types and the problems they face everyday. Moreover, in this chapter we will discuss how can we use the technologies of the semantic web of things in the same context. This concept combine between The Internet of Things and The Semantic Web. We will talk about their architectures and challenges.

1.2 Persons with disability

According to the definition of the Health Organization (WHO), a disabled person is any person whose physical or mental integrity is temporarily or permanently impaired, either due to age or accident, so that his or her autonomy and the ability to do daily tasks of normal persons is compromised.[World Health Organization, 2020]

1.2.1 Types

The different types of disabled persons is shown below : [Alpes-Maritimes, 2016]

- Vision Impairment.
- Deaf or hard of hearing.
- Mental health conditions.
- Intellectual disability.
- Acquired brain injury.
- Autism spectrum disorder.
- Physical disability.

1.2.2 Common Problems encountered by the persons with disabilities

The difficulties encountered by handicaps are the following:

- The difficulty of doing daily life's tasks such as clothing, hygiene, food, activities at home, etc.
- Social life such as communication with others.
- Relocation and moving.
- Studying at school or university as well as finding a job.
- The lack of respect and consideration, the underestimation as well as the aggressiveness applied to people with disabilities by other normal people.

1.2.3 Solutions

With the evolution of science and technology, the problems of the persons with disabilities can be resolved partially at least. In the next section we are going to discuss two disciplines that we are going to use to resolve those problems and make the life of these kind of persons easier. The two disciplines are Semantic Web and Internet of Things.

1.3 Internet Of Things

Internet of things is basically a new field which made it hard to stick with a standardized definition, so we picked the most convenient definition to us. According to The Coordination and Support Action for Global RFID-related Activities and Standardisation : "A global network infrastructure, linking physical and virtual objects through the exploitation of data capture and communication capabilities. This infrastructure includes existing and evolving Internet and network developments. It will offer specific object-identification, sensor and connection capability as the basis for the development of independent cooperative services and applications. These will be characterised by a high degree of autonomous data capture, event transfer, network connectivity and interoperability."[Casagras, 2009]

The data transmission between smart things has been studied for decades. Diverse technologies and standards have been proposed in this field. Making the smart things inter-connectable such that bits can be transferred between devices, and this is only the beginning, more works are expected to make smart devices coherent which means they understand each other. Interoperability is particularly essential, and a must, to build system with various devices, especially those from different manufacturers. The web was used to connect those different devices due to its unified protocols which will improve the interoperability, We call it Web of Things.[Chatzimichail *et al.*, 2021]

1.3.1 Architecture of IoT

We will present the different layers of an IoT architecture bellow: [Zhong *et al.*, 2015]

- **Devices Layer:** The IoT device layer includes of all the smart devices that are connected to the system. Those devices are materials that are embedded with sensors, processors, actuators, and the capability of transmitting data over the networks.
- **Gateway Layer:** The gateway layer is the middleware between the IoT device layer and the IoT platform layer. This layer contains a physical device or software program that gather data from intelligent devices and transfer it to the cloud.
- **Platform Layer:** After transmitting data to the cloud, it can be processed by tools and applications in the IoT platform layer. Which also includes software that visualize processed sensor data on user-facing devices.

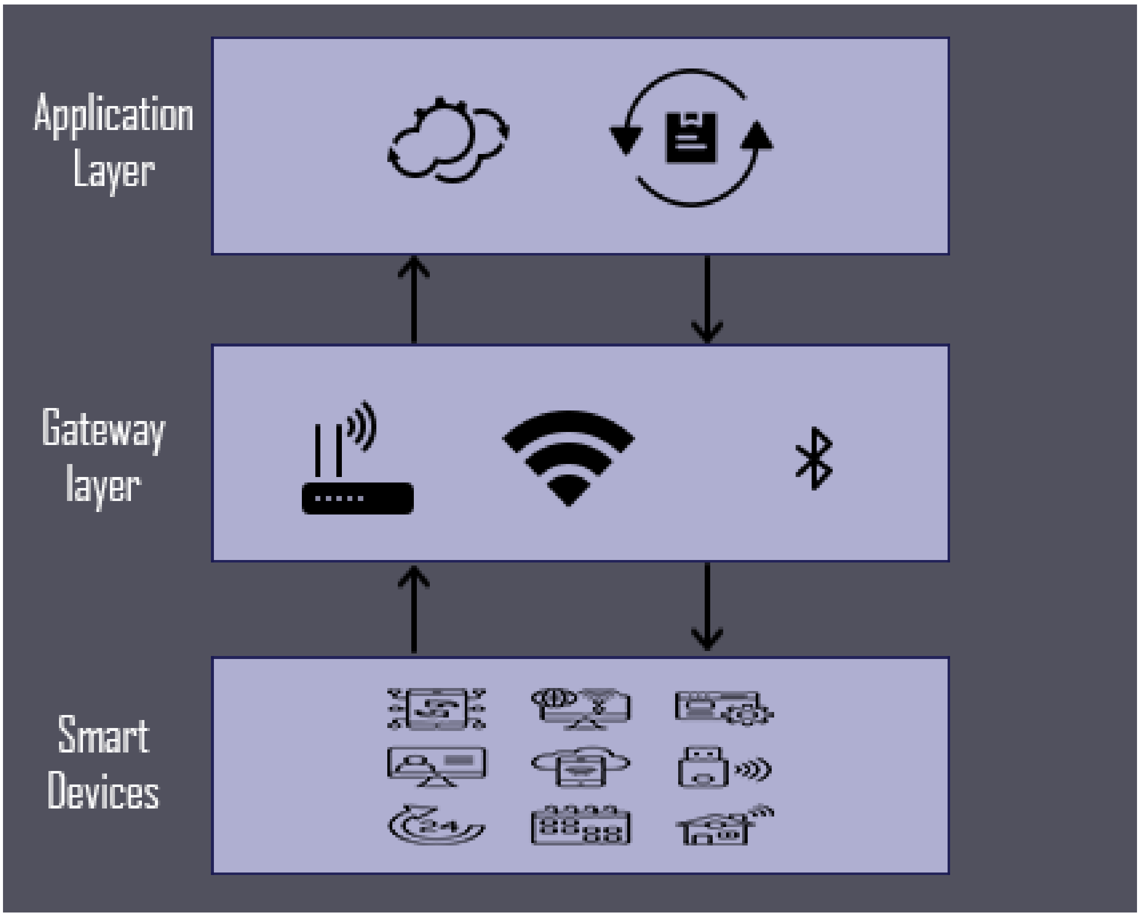


Figure 1.1: Architecture of IoT

1.3.2 Characteristics of IoT

According to the article of i-SCOOP there is 7 characteristics of IoT, We are going to list and explain [i-SCOOP, 2022]

- **Connectivity :** Due to the infrastructure of IoT, every thing can be connected and inter-communicated.

- **Things:** Anything that can be connected such as phones, computers, cameras, and drones, etc. Devices can contain sensors or sensing materials can be attached to devices and items.

- **Data:** Data is the main component of the Internet of Things, the first step towards action and intelligence.

- **Communication:** Things get connected which allow them to communicate and share data. This data can be analyzed. The communication can be established over the different types of networks: LAN, MAN and WAN.

- **Intelligence:** The aspect of intelligence as in the sensing capabilities in IoT devices and the intelligence gathered from big data analytics and processing.

- **Action:** The consequence of intelligence. This can be done manually or automated by software implemented in the devices.

- **Ecosystem:** The Internet of Things from a perspective of other technologies, communities, goals and the place in which the Internet of Things fits. The IoT can function as a unit with multiple technologies and can be used in different fields.

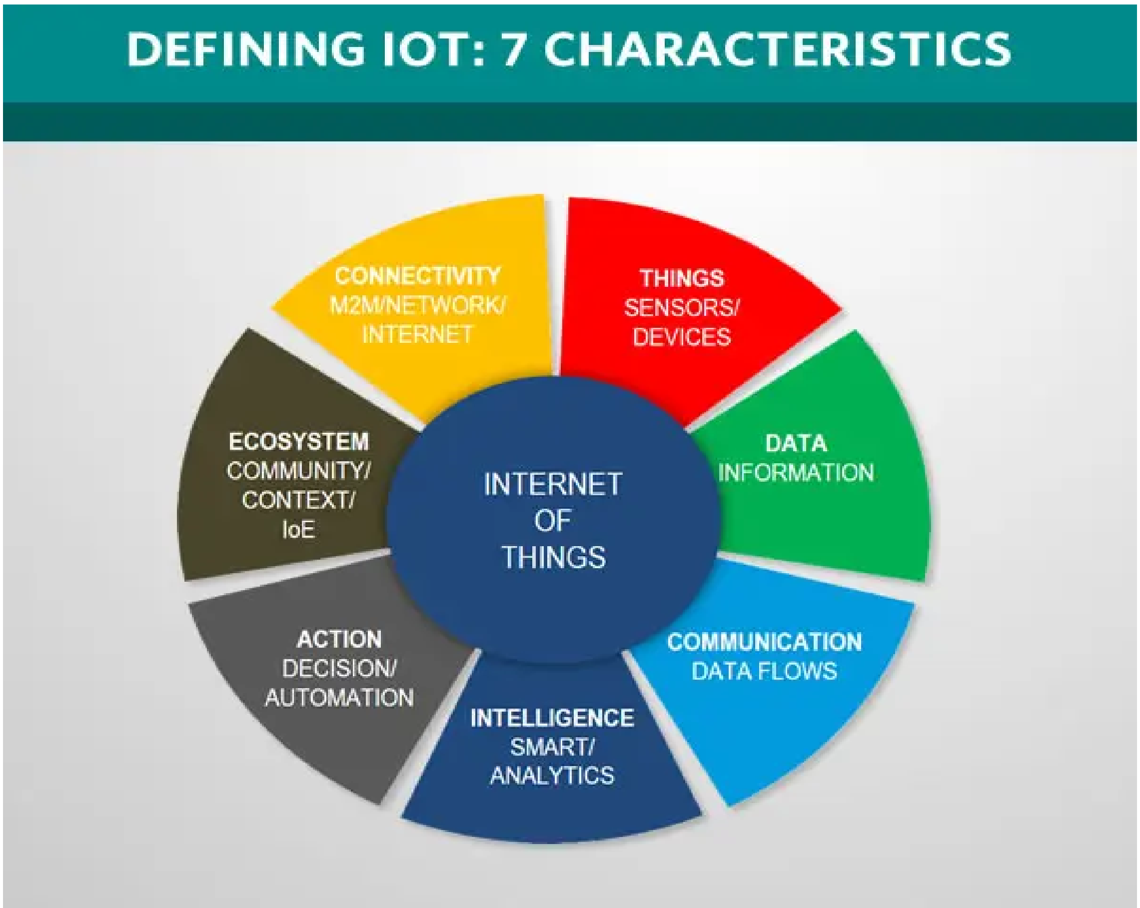


Figure 1.2: Characteristics of IoT [i-SCOOP, 2022]

1.3.3 Challenge

Today there are a lot of challenges that faces the Internet of things, but the most important challenge that we are concerned with is the interoperability. It can be defined as the ability to make two or more systems/components communicate by exchanging data with each other and by using in and out of the box tools. Or by using information. This is why we call upon to the semantic web in the next section.

1.4 Semantic Web

Semantic web is a set of technologies that aims to better describe resources to make them more understandable for the machine due to a formal metadata system. It allows the actual web to exploits all its content. Let's imagine all of the data existing in the world wide web linked together with meanings that allows the machine to understand and act based on it.

1.4.1 Architecture of Semantic Web

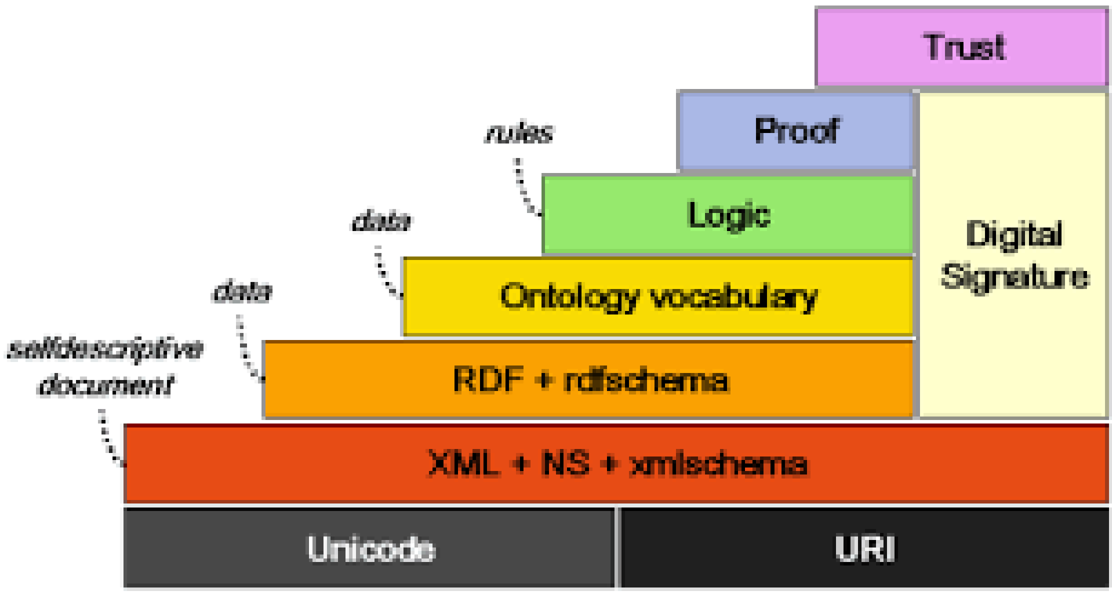


Figure 1.3: Semantic Web Architecture [Berners-Lee, 1998]

1. **Unicode and URI:**

   Unicode and URI (uniform resource identifiers) layers are the first ones of the Semantic Web Architecture; they enforce the use of international characters, they identify resources on the Web such as documents, images, downloadable files and services.

2. **XML, XML NS, and XML Schema:**

   The layer constituted of XML, XML Name Space and XML schema allows a uniform structure representation for documents, while XML is a markup Language created to provide us with generalities across the internet due to its simple format which is readable for both Human and machine. In addition, the XML name space is used to provide us with a method to avoid element name conflicts and allows the combination of documents with heterogeneous vocabulary. Moreover, XML Schema describes the structure of and XML document.

3. **RDF and RDFS:**

   The next layer is RDF and RDFS: by using those technologies, it is possible to link web ressources with a pre-defined vocabulary. First, Resource Description Framework (RDF) is a directed graph contains simple three-part data model (subject, predicate, object) for referring to objects or resources and how they relate to each other. Also, RDF Schema (RDF-S) describes the properties, classes, and hierarchies of RDF resources.

4. **Web Ontology Language (OWL):**

   OWL is used in the ontology layer to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. This representation of terms and their inter-relationships is called Ontology. OWL has more facilities for expressing meaning than XML, RDF and RDF-S, and it goes beyond those languages in its ability to represent machine-interpretable content.

5. **Logic and Proof Layers:**

   Finally, the logic and proof layers allow the definition of formal rules and the reasoning based on them.

1.4.2 Ontology:

Ontology is the term used to refer to the mutual understanding of some domain of interest which may be used as a unifying system, an Ontology necessarily represents some sort of world perspective with respect to a given field.[Uschold & Gruninger, 1996]

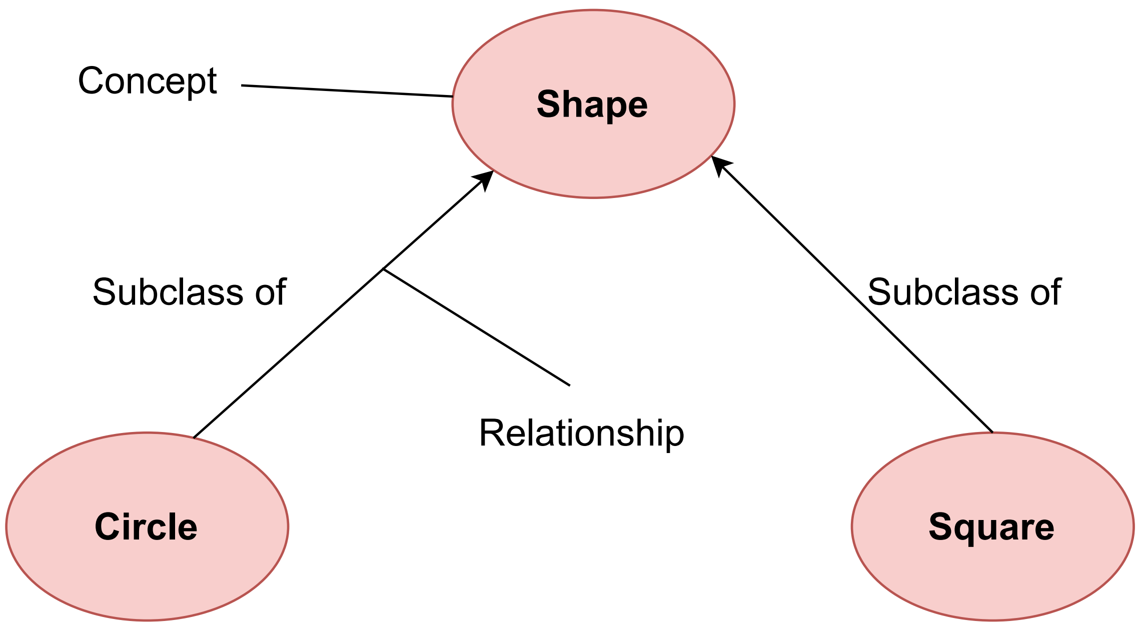


Figure 1.4: Ontology Example

Components of ontology

Things we can use to represent something using Ontology:

1. **Concept:** Concepts can be abstract or concrete, elementary or composite, real or fictitious, in short, a concept can be anything , Concepts are also known as classes (XOL, RDF(S), OIL, DAML+OIL), objects (OML) or categories (SHOE). for example : Person, City, Sensor, etc.

2. **Instances:** Instances represent elements related to a specific concept. Oussama for example is an instance of the concept person . Guelma is an instance of the concept City, etc.

3. **Relations:** Relations are an interaction between concepts of domain and range. They are called relations in SHOE and OML, and roles in OIL.

4. **Attributes:** An inherent characteristic of a concept that better defines it. Every attribute has a value that differs from an instance to another. For example: name of the concept Person has the value of BADJI.

5. **Axioms:** Axioms are statements that are asserted to be true. For example: we have two concepts, Person and Brother, our statement is: the brother is a person.

1.4.3 Technologies used in semantic web

1. **RDF:** The Semantic Web uses RDF as a format to store information. It is by nature open-ended. Which means that we can always add new data or relationships. Another advantage of using RDF is that it supports unstructured data, documents, websites, articles and more.

2. **OWL:** is a language designed to represent complex knowledge about things and the relations between them, This language offers the descriptive feature to the semantic web which means that ontologies are independent of the data that they describe. However, in classic databases, the data described are determined directly.

3. **SPARQL:** While RDF and OWL allow to store and define knowledge and relationships. It is necessary to have a query language to take advantage of them. SPARQL is a protocol, and RDF's query language [Prudhommeaux & Seaborne, 2008] . And also we can see it like an SQL for the Semantic Web. It uses the principle of identifying paths in a graph to retrieve the results of a given query. One of the advantages of SPARQL is that queries can be distributed to multiple SPARQL endpoints (services that accept SPARQL queries and return results).

4. **SWRL:** Semantic Web Rule Language [Lapique, 2006] is a language of the logical layer in the SW architecture that enhance the semantics of a defined Ontology in OWL. It allows to manipulate instances by variables (?x,?y,?z). However, SWRL does not allow to create concepts nor relations, it simply allows to add relations according to the values of the of the variables and the satisfaction of the rule. Its rules are built following on this scheme: antecedent -> consequent.

1.4.4 Application fields

Semantic web is applied in multiple fields such as the following:

- **Web Search and E-commerce:** Search engines genuinely have the benefit of having access to further metadata so as to come back with additional relevant results. In fact, the most important players in the industry are investing heavily in standards that encourage firms to annotate their web pages with considerably additional structure, which was one of the original purpose of the Semantic Web vision in the first place. A lot of companies developed its technologies such as OGP (Open Graph Protocol) created by Facebook, which is very similar to RDF. Microsoft, Google, and Yahoo use Schema.org, which contains an RDFa representation. The E-commerce Industry has GoodRelations, which also uses RDFa. These frameworks are all currently used to bring users a better web experience.

  In addition another excellent case study on the usage of Semantic Web technologies is Best Buy. They adopted GoodRelations for their website and saw a fantastic increase in hits and conversions. Jay Myers has presented at numerous conferences, and his work on the subject can be found everywhere in the web. [Cambridge Semantics, 2021]

- **Healthcare:** Nowadays, wherever you go, the healthcare services are indispensable. To provide those services in an optimal way, information exchange is a must especially in the medical sciences since much of the medical information that is available needs a way to be shared disparate computer systems. Ontology-based inter-operation has been intended for sharing knowledge and exchanging information both across people and across services/applications, using multiple connected devices from different manufacturers that uses different standards which will be treated semantically using the semantic web. [Karami & Rahimi, 2019]

1.4.5 From the Internet of Things towards the Semantic Web of Things

Recent research on semantic web has shown the power of semantic to manage structured, semi-structured, and unstructured data. Which allow us to benefit from Big Data collected from various data sources (IoT devices) trying to get better results. This combination still in early stages of research which is a challenge that will contribute to science in different fields such as health care domain, business, and financial domain. In the next subsections we will show how we solved the problem of interoperability related to IoT and how we can benefit from it to help disabled persons going from Internet of objects to the Web of Things and finally to the Semantic Web of Things. [Gacitua *et al.*, 2018]

1.4.6 From the Internet of Things to the Web of Things

As more and more smart devices are getting connected to the Internet, the best step to take is to use World Wide Web and its technologies and protocols to make those devices inter-operable like we said in the previous subsection. Few years ago, Kindberg et al proposed to link physical objects with web pages using infrared interfaces or bar codes on objects. Users could find the URI of the related page simply by interacting with the object.[Kindberg *et al.*, 2002]

1.4.7 Semantic Web of Things (SWoT)

While the previous sections outline the benefits of integrating web of things, this section dives deeper into integrating semantics to WoT. It explains the way to semantically interpret data produced by IoT devices by using the W3C Semantic. So what is a Semantic Web of Things ?

Definition

The Semantic Web of Things (SWoT) is a kind of extension of the W3(World Wide Web) trying to solve the problems caused by heterogeneous systems and information and to provide a better understanding of the different domains of the IoT. It can also be described simply as a combination of the Semantic Web (WS) and the Internet of Things (IoT). This approach allows abstractions that are appropriate for matching sensors with their raw data to exist. [Chatzimichail *et al.*, 2021]

1.4.8 Features of combining SW with IoT

Researchers have combined the characteristics that SW and IoT has, including the following:

- Generalized use of URIs and the HTTP protocol
- Enabling interoperability among data from different sources through the content annotation.
- Use of common standard languages.
- Domain knowledge and background information combination with sensor data, adding more meanings to machines. [Chatzimichail *et al.*, 2021]

1.4.9 Challenges of SWoT

Common challenges of Semantic Web of Things encountred by researchers :

- The scalability and growth of IoT systems with multiple devices.
- The lack of developers interested by this field.
- The lack of semantic data sets.
- The precision of reasoning my not be optimal.

1.5 Conclusion

At the end, we have presented in this chapter the most related concepts to handicaps health needs and the technologies provided to help them become more autonomous and benefit from easy daily life. The technologies based on IoT that is used to gather medical data from patients, processing it, and then exchange it with others. The chapter also explained the IoT challenges, especially heterogeneous data and the Semantic Web as a solution used to deal with it. In the following chapter we will present other concepts that will be applied in this work to optimize the semantic system without loosing its precision.

Chapter 2

Machine Learning

2.1 Introduction

As mentioned in the first chapter, Semantic Web give us the ability to handle the heterogeneity of data collected from various devices. This started a challenge for researchers to optimize the actual solutions to have less processing time and resource consumption. Therefore we intend to use a machine learning algorithm, which we will discuss in this chapter. We went through using data from heterogeneous devices and process them using Semantic Web.

2.2 Definition

Machine learning is a way to simulate human learning for computers, and teach the machine how to self-improve for getting new knowledge and skills. Compared with human learning, machine learning learns faster, and without a limitation of how much information can be learned. the gathering of knowledge is more easy, the results of learning spread easier. So, any achievements of the humans in the field of machine learning will escalate the capability of computers, thus have an impact on human society.[Wang *et al.*, 2009]

2.3 Types of ML

Machine Learning is a very wide field that we can represent in 3 main types:

- **Supervised Learning:** Supervised machine learning needs an input data (historical data of input-output examples). Then, the algorithm will predict or classify new information based on it.
- **Unsupervised Learning:** Unsupervised machine learning is referred as class discovery. One of the major differences between unsupervised and supervised machine learning is that no training data is required for the unsupervised.
- **Reinforcement Learning:** Reinforcement learning is different from unsupervised learning in terms of goals. The goal in the unsupervised learning is to find similarities and differences between data-points, while in the reinforcement learning problem, the goal is to find a good behaviour, an action or a label for each particular situation if you are willing to maximize the long-term results.

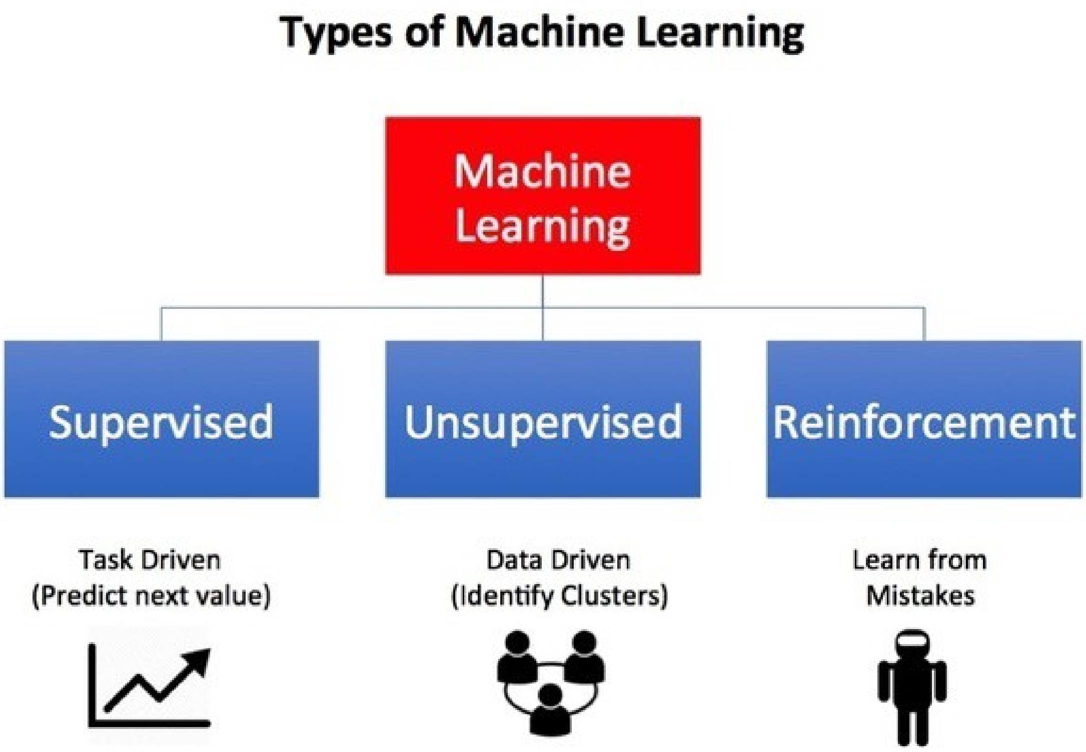


Figure 2.1: Types of ML

2.4 Why optimizing ?

Before Talking about the optimization process, we must be clear about our goals when we say we want to optimize a function. What are we trying to achieve when we optimize? The optimization look for improving performance to reach an optimal solution: an accurate decision for a handicap person in the shortest possible time, because the small possibility of faults and the late decisions in the healthcare field can cause a disaster.

2.5 Optimization Algorithms in ML

1. **Backpropagation** The Backpropagation algorithm is a supervised learning method for multilayer feed-forward networks from the field of Artificial Neural Networks. [Zajmi *et al.*, 2018]

2. **Gradient Descent** Gradient descent (GD) is an iterative first-order optimisation algorithm used to find a local minimum/maximum of a given function. This method is commonly used in machine learning (ML) and deep learning(DL) to minimise a cost/loss function (e.g. in a linear regression). Due to its importance and ease of implementation.

3. **Naive Bayes optimization** Bayesian Optimization is an approach that uses Bayes Theorem to direct the search in order to find the minimum or maximum of an objective

4. **Genetic Algorithms** Genetic algorithms are search algorithms based on the natural human genetics.

Characterestics Algorithms	**Backpropagation**	**Gradient Descent**	**Gentics Algorithm**	**Native Bayes**
Speed	Depending on number of layers more layers means slower execution	Depending on the type of Gradient Descent Stochastic GD > Mini-batch GD > Batch GD	Depending on the number of iterations	Slow
Complexity	Yes	Yes	Yes	Yes
Flexibility	Flexible	/	Flexible	Flexible
Area of application	Neural Networks	Neural Networks	Any Optimization problem	Any Optimization problem

Table 2.1: Optimization Algorithms Comparison.

2.5.1 Genetic Algorithms

Genetic algorithms are search algorithms created for the sake of optimization. They combine the results of fitness function execution to select the best genes, the fitness function is a function which takes a candidate solution to the problem as input and produces as output how "good" the solution is with respect to the problem in consideration and the optimal solution we are looking for. In every produced generation, a new set of artificial children is created using bits and pieces of the old results of fitness function. While the first iteration is randomized, genetic algorithms are not fully random. They efficiently exploit historical data to estimate on new search points with expected improved performance. These algorithms have been developed by John Holland, his colleagues, and his students at the University of Michigan. [Goldberg, 1989]

2.5.2 GA Architecture

A typical cycle of genetic algorithm basically starts from initial selection via crossover, to mutation. If the result does not fit the objective results, the algorithm will loop using the new offspring as the initial population of chromosomes[Jing, 2016]. The Figure below shows the logic that how GAs works:

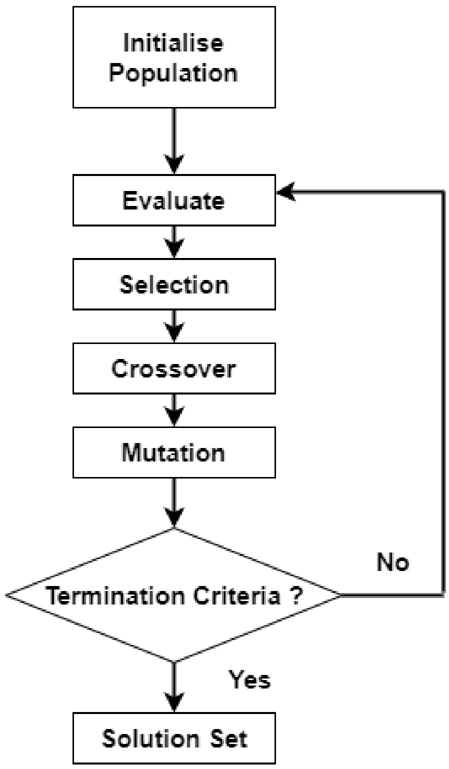


Figure 2.2: Basic Architecture of Genetic Algorithm

2.5.3 GA Components

Genetic Algorithms are composed mainly of 3 things consists of : [Roetzel *et al.*, 2020]

1. **Reproduction :** Reproduction is a process based on the fitness function of each gene. This function identifies how much a gene is good. Moreover, genes with

better fitness result have higher probability of contributing offsprings to the next generation.

2. **Crossover :** Crossover is a process in which members of the last population are paired at random in a pairing pool. So, a pair of offsprings is generated, merging elements from two chromosomes, which hopefully have improved their fitness values.

3. **Mutation :** Mutation is an infrequent process (with small probability) which consists of random alteration of the value of a gene position. Actually, mutation is a process of random walk through the coded parameter space. Its purpose is to ensure that important information contained within genes may not be lost too early.

2.6 Related Works

- Authors in [Takabayashi *et al.*, 2018] have evaluated preamble detection in the ETSI SmartBAN PHY and proposed a modified preamble structure. Specifically, an SFD was added by them between the two-octet preamble and the PLCP header. The proposed preamble structure is compatible with the SmartBAN standard.

- Haghi Kashani a, Mona Madanipour b, Mohammad Nikravan a, Parvaneh Asghari c, Ebrahim Mahdipourin in there article “A systematic review of IoT in healthcare: Applications, techniques, and trends Mostafa” they present a comprehensive taxonomy in the HIoT, analyze the articles technically, and classify them into five categories, including sensorbased, resource-based, communication-based, application-based, and security-based approaches. Furthermore.[Haghi Kashani *et al.*, 2021]

- Amir Masoud Rahmani, Zahra Babaei and Alireza Souri in there review “Event-driven IoT architecture for data analysis of reliable healthcare application using complex event processing”, In this paper, an event-driven IoT architecture is presented for data analysis of reliable healthcare applications, including context, event, and service layers ... etc.Moreover, Dependability parameters are considered in each layer, and the CEP method as a novel solution and automated intelligence is applied in the event layer. The implementation results in this paper have showed that the CEP method increased reliability, reduced costs, and improved healthcare quality.[Rahmani *et al.*, 2020]

- Akila D and Balaganesh in their article “Semantic Web-based critical Healthcare system using Bayesian" networks” they designed Semantic Web-based Critical Healthcare (SWCH) system for early early diagnosis and treatment of critical diseases which include Heart attack, Stroke, Cancer, Kidney failure and Brain tumour.It consists of knowledge Extraction and diagnosis and disease prediction modules. In Knowledge extraction module, medical data from various sources is gathered. Based on this knowledge base, a medical Ontology model with Bayesian networks is developed which provides the diagnosis of whether the patient has the potential to have that critical disease or not.[Akila & Balaganesh, 2021]

- Rashmi Burse, Michela Bertolotto, Dympna O’Sullivan, and Gavin McArdle in their paper “Semantic interoperability: the future of healthcare", They discussed the interoperability in healthcare and how it is a big challenge in this field also they reviewed the various healthcare standards developed in an attempt to solve the interoperability problem at a syntactic level and then moves on to examine medical ontologies developed to solve the problem at a semantic level.[Burse *et al.*, 2021]

- Sapna Juneja,Abhinav Juneja,Annu Dhankhar, and Vishal Jain in their chapter 8 of "Semantic Web for Effective Healthcare systems" book, They shed the light on the growth of the IoT technology and its role in the development of the medical field as it becomes very easy for the patients to get connected to the doctors and healthcare professionals at any time virtually. In this research work, the authors defined the applications of IoT in healthcare. moreover, they mentioned how these applications can be used with the help of various sensors. And how the data that has been collected from these sensors and shared on the cloud for the storage purpose. [Jain *et al.*, 2021]

- Alexander Hogenboom, Viorel Milea, Flavius Frasincar and Uzay Kaymak in their conference paper with the title "RCQ-GA: RDF Chain Query Optimization Using Genetic Algorithms", they focused on optimizing a special class of SPARQL queries, the so-called RDF chain queries. For this purpose, they devised a genetic algorithm called RCQ-GA that determines the order in which joins need to be performed for an efficient evaluation of RDF chain queries. Their approach is benchmarked against a two-phase optimization algorithm and it outperformed them.[Hogenboom *et al.*, 2009]

2.7 Discussion

Several works were cited in the previous section about IoT, Semantic Web of Thing, and Optimization algorithms in the healthcare field. Semantic Web of things showed a great improvement and contribution in the field of the healthcare which inspired us to try it on a micro field: Persons with disabilities, and after reading previous work we found that its limitation is mainly concentrated in reasoning processing time and the high resource consumption. This why we will propose an approach that combines the use of semantic Web and Genetic Algorithms to solve these problems.

2.8 Conclusion

At the end, in this chapter, we presented multiple technical methods that can help people with disabilities in their daily life, and we mentioned some previous works in this field. We believe that our proposed approach will to some extent overcome the limitations of those previous works. In the next chapter, we will explain our method and provide more details.

Chapter 3

Conception

3.1 Introduction

Healthcare field is very sensitive, a small wrong decision can cause a disaster, but it can benefit from technologies like Internet of Things to gather medical data, and semantic web to interpret those information to something meaningful by adding metadata to resources that will allow an automatic system to make decisions like service recommendation. The main idea of semantic web is to make content understandable by machines, the content of medical data can be combined with specific concepts, relations, and rules in order to infer new knowledge that adapts with our exact context. Semantic Web seems to improve the precision of healthcare decisions. However, with the development of the world and the appearance of the concept Big Data produced a big problem that consists of slow processing of data and of course hardware resources consumption due to the large amount of information and reasoning that transform these information into meaningful concepts. In this thesis, we intend to use genetic algorithm to improve our system and reduce the amount of rules for a faster processing of information.

3.2 Objective of proposed method

To increase the functional capabilities of disabled and elderly people, several computing devices have been designed. The Internet of Things (IoT) is one of the recent technological developments that can contribute greatly in the care of this population. Technological advances and access to the Internet over the past decade have made it possible to connect everyday objects to the Internet. Our objective is to use this technology to collect data from handicap persons. Moreover, this data can be very huge due to the high number of sensors that continually collects information that needs to be processed and manipulated. It is a must to use semantic web combined with the IoT to provide data in a meaningful and efficient way to achieve interoperability. Semantic processing can take too much time and resources for processing this data and recommend a service based on it to help the people with disability. An intelligent optimization algorithm needs to be combined with the previous two technologies to achieve this objective which we will discuss in the next sections of this chapter with our proposed approach.

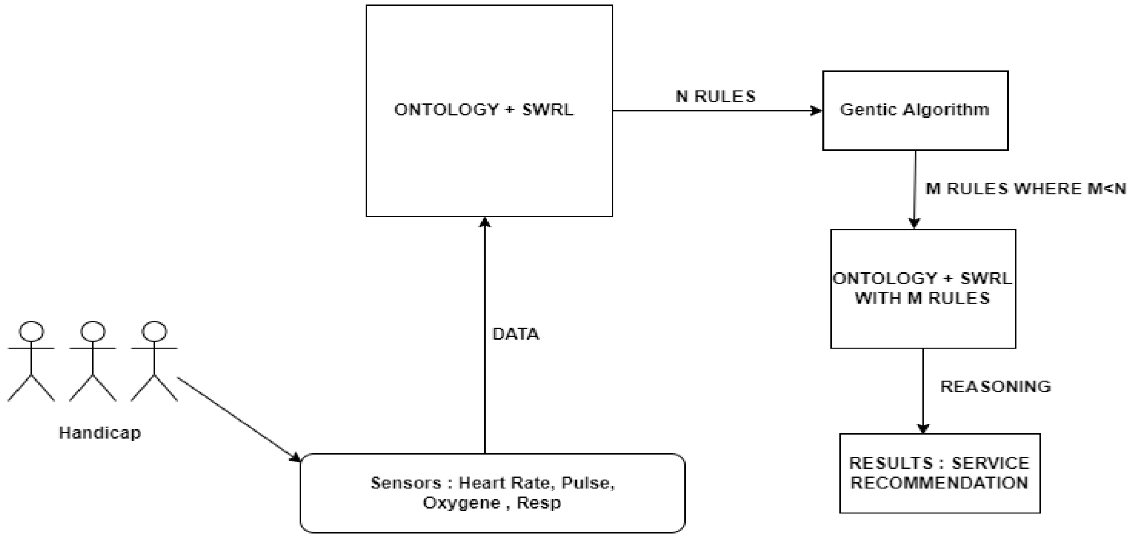


Figure 3.1: Architecture of proposed system

3.3 Architecture of the proposed system

Explanation The system starts by collecting useful information from handicaps using sensors such as vital signs sensors. Every sensor can has its own manufacturer and measurement unit. Using ontology, the interoperability is guaranteed because it can understands the different representation of a knowledge about those sensors. Based on a reasoning engine, we used SWRL rules to infer new knowledge. These rules can be reduced without loosing accuracy of the system using a machine learning optimization algorithm, which is the genetic algorithm, that returns a set of rules. We run reasoning again with the new reduced rules and we maintain the same accuracy rate, so that the perfect service is recommended to the handicap by our system.

3.4 Presentation of knowledge

The usage of technologies of semantic web in the healthcare field, enabled us to recommend services to a handicap patient based on his vital signs. In the following sections, we will discuss how did we use the semantic web model to describe the knowledge of a handicap environment.

3.5 The ontology

In our work context, we created an ontology that better describes healthcare field related to a handicap patient, we tried to focus on tracking the handicap vital signs to help him dealing with his actual situation. The lack of information and data sets in the semantic field was a big challenge for us which limited our ontology to depend only on vital signs to make the decision.

3.5.1 Concepts and classes

1. **Person :**
    - Handicap_person
    - Doctor
    - Assistant
2. **Assistant_technology**
    - Wear_Glasses
    - Wheel_chair
3. **Location**
4. **Measurement**
    - hr
    - oxygene
    - pulse
    - resp
5. **Medical_Background**
    - Breathing_Illness
    - Diabetic_Illness
    - Heart_Illness
    - Low_Vision_Illness
6. **Objectif**
    - Travel
    - Shopping
    - Go_To_Work
7. **Obstacle**
    - Bad_Weather
    - Closed_road
8. **Sensor**
    - Body_Tempurature_sensor
    - Camera
    - GPS
    - HR_sensor
    - Oxygene_sensor
    - Pulse_sensor

- RESP_Sensor

9. **Service**
   - Call_doctor
   - Drug
   - Hospitalize
   - Food
   - Take_drug
10. **Type_of_disability**
    - Vision_impairement
    - Deaf
    - Mental
    - Physical
11. **Weather**
    - Rainy
    - Sunny

Concepts	**Description**
Person	The concept person represents a human. It regroups Doctors, Assistants, And Handicap patients.
Handicap_person	Handicap person represents a person that has a disability regardless its type.
Doctor	Doctor represents a person that can treat handicap patients.
Assistant	The Assistant represents a person that can help handicap patients depending on their needs or doctor instructions.
Assistant_Technology	Represents a set of tools that can assist an handicap person.
Glasses	Glasses represents a tool that can help a person to improve his vision.
Wheel chair	Wheel chair represents a tool that can help a person to move towards an objective location.
Location	A location represents a place in the map defined by coordinates.

Measurement	The concept measurement represents a number gathered from sensors. It heart rate, oxygen, pulse, and respiratory rate.
hr	Hr represents the value gathered from heart rate sensor.
oxygene	Hr represents the value gathered from heart rate sensor.
pulse	pulse represents the value gathered from pulse sensor.
resp	RESP represents the value gathered from RESP sensor which is the respiratory rate.
Medical_Background	Medical background represents the medical history of an handicap patient.
Breathing_Illness	Breathing Illness is a subclass of medical background and it represents if an handicap patient has a breathing Illness in his history.
Diabetic_Illness	Diabetic Illness is a subclass of medical background and it represents if an handicap patient has a Diabetic Illness in his history.
Heart_Illness	Heart Illness is a subclass of medical background and it represents if an handicap patient has a Heart medical problem in his history.
Low_Vision_Illness	Low Vision Illness is a subclass of medical background and it represents if an handicap patient has a vision problem in his history.
Objectif	The objectif concept represents a task that the handicap wants to accomplish.
Obstacle	The Obstacle concept represents something that interrupt the handicap while he is doing an objective task.
Sensor	The sensor class represents a device that is connected to the handicap patient and the semantic system, It gathers information and transfer it to the system.

Body_Temp_sensor	The Body Tempurature sensor measure the body tempurature of the handicap and it is a subclass of sensor.
Camera	The Camera concept represents a device that captures images and video to detect obstacles.
GPS	The GPS concept represents a device that detect actual position of the handicap.
Oxyegene_sensor	The Oxugene sensor represents a device that measure the amount of o2 in blood.
Pulse_sensor	The pulse sensor represents a device that measure the blood flow.
HR_sensor	The HR sensor represents a device that measure heart rate of the handicap.
RESP_sensor	The respiratory sensor is a device that measure the respiratory rate of the handicap person.
Service	Service is a concept that represents something to do for the handicap and it regroups a set of services such as taking drug and hospitalization ...etc.
Call_doctor	It is a service that is suggested to the handicap based on his health condition.
Take_drug	It is a service that is suggested to the handicap based on his health condition.
Hospitalize	It is a service that is suggested to the handicap based on his health condition.
Food	It is a service that is suggested to the handicap based on his health condition.
Type_of_disability	The Type of disability concept represents the limitation of the person.
Weather	The Weather concept represents the condition of the weather and the temperature.

Table 3.1: List of concepts

3.5.2 Relationships between objects

Concepts alone can't describe the knowledge we need in our system. This is why we need to add proprieties and relationships that better describe those objects and how they are manipulated in the system, The next table contains all the proprieties that describe relationships between objects

Object Propriety	**Domain**	**Range**	**Function**	**Description**
Benifit_from_doc_call	Handicap_patient	Call_doctor	/	The propriety represents the relation benifit from doctor call connects the handicap patient with service call doctor.
Benifit_from_drug	Handicap_patient	Drug	/	The propriety represents the relation benifit from drug connects the handicap patient with service drug.
Benifit_from_food	Handicap_patient	Food	/	The propriety represents the relation benifit from food connects the handicap patient with service food.

Benifit_from _hospitalization	Handicap_patient	Hospital	/	The propriety represents the relation benifit from hospitalization connects the handicap patient with service hospital
canHelp	Person Assistant	Person Handicap_patient	needs	The propriety represents the relation can help connects the assistant with the handicap patient describes the action help.
Has_hr_meas	Sensor HR_sensor	hr Measurement	/	The propriety represents the relation Has hr measurement connects the sensor with the measurement of heart rate.
has_hr_sensor	Person Handicap_patient	Sensor HR_sensor	/	The propriety represents the relation Has hr sensor connects the handicap with the heart rate sensor.

Has_oxygene_meas	Sensor Oxygene_sensor	oxygene Measurement	/	The propriety represents the relation Has oxygene measurement connects the sensor with the measurement of oxygene.
has_oxygene_sensor	Person Handicap_patient	Sensor Oxygene_sensor	/	The propriety represents the relation Has oxygene sensor connects the handicap with the oxygene sensor.
has_pulse_meas	Sensor Pulse_sensor	pulse Measurement	/	The propriety represents the relation Has pulse measurement connects the sensor with the measurement of pulse.
has_pulse_sensor	Person Handicap_patient	Sensor Pulse_sensor	/	The propriety represents the relation Has pulse sensor connects the handicap with the pulse sensor.

Has_resp_meas	Sensor RESP_Sensor	resp Measurement	/	The propriety represents the relation Has respiratory measurement connects the sensor with the measurement of respiratory rate.
Has_resp_sensor	Person Handicap_patient	Sensor RESP_Sensor	/	The propriety represents the relation Has respiratory sensor connects the handicap with the respiratory sensor.
needs	Person Handicap_patient	Person Assistant	canHelp	The propriety represents the relation needs that connects the handicap patient with the assistant describes the action needs.
treatedBy	Person Handicap_patient	Person Doctor	treats	The propriety represents the relation treated by that connects the handicap patient with the doctor describes the action treated by.

treats	Person Doctor	Person Handicap_patient	treatedBy	The propriety represents the relation treats that connects the doctor with the handicap patient describes the action treats.

Table 3.2: List of object proprieties.

3.5.3 Literal proprieties

Data Propriety	**Domain**	**Range**	**Description**
Age	Person Doctor Handicap_patient Assistant	Integer	Represents a number that describe the age of person
Condition	Weather	String	Represent a description of the weather condition.
First_Name	Person Doctor Handicap_patient Assistant	String	Represent the given name after birth to a person.
Name	Person Doctor Handicap_patient Assistant	String	Represent the family name that all close sibling have.
Sexe	Person Doctor Handicap_patient Assistant	String	Represent the sexual description of the person.

Temperature	Weather	Float	A number that represents the measurement of how cold or hot is the weather.
time	Measurement	Integer	Represent a number that describe a moment of time.
value	Meauserement	Integer	Represents a number which describe a measurement.

Table 3.3: List of data proprieties

3.5.4 SWRL Rules

We added SWRL rules to our system, the ontology can be greatly enhanced in completeness, expressiveness, and logic. This is because SWRL allows users to specify user-defined rules in order to perform inferences over OWL individuals, so new knowledge about these individuals can be acquired. We enriched our system with a set of rules chosen by a doctor, we will explain each one of them in the next lines:

1. **Rule 1:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : respiratory < 12 and oxygen < 96 and 54 < heart rate < 100 then this handicap person will benefit from hospitalization service

   Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:lessThan(?v, 12) ŝwrlb:lessThan(?vo, 96) ŝwrlb:greaterThan(?vh, 54) ŝwrlb:lessThan(?vh, 100) Ĥospitalize(?ser) -> Benifit_from_hospitalization(?h, ?ser)

2. **Rule 2:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : respiratory < 12 and oxygen > 95 and heart rate > 150 then this handicap person will benefit from hospitalization service

   Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:lessThan(?v, 12) ŝwrlb:greaterThan(?vo, 95) ŝwrlb:greaterThan(?vh, 150) Ĥospitalize(?ser) -> Benifit_from_hospitalization(?h, ?ser)

3. **Rule 3:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : respiratory > 21 and oxygen < 96 and heart rate > 150 then this handicap person will benefit from hospitalization

service

Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:greaterThan(?v, 21) ŝwrlb:lessThan(?vo, 96) ŝwrlb:greaterThan(?vh, 150) Ĥospitalize(?ser) -> Benifit_from_hospitalization(?h, ?ser)

4. **Rule 4:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : respiratory $>$ 21 and oxygen $>$ 95 and heart rate $>$ 150 then this handicap person will benefit from hospitalization service.

   Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:greaterThan(?v, 21) ŝwrlb:greaterThan(?vo, 95) ŝwrlb:greaterThan(?vh, 150) Ĥospitalize(?ser) -> Benifit_from_hospitalization(?h, ?ser)

5. **Rule 5:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : respiratory $>$ 21 and oxygen $>$ 95 and heart rate $>$ 100 then this handicap person will benefit from drug.

   Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:greaterThan(?v, 21) ŝwrlb:greaterThan(?vo, 95) ŝwrlb:greaterThan(?vh, 100) T̂ake_drug(?ser) -> Benifit_from_drug(?h, ?ser)

6. **Rule 6:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : respiratory $>$ 21 and oxygen $<$ 96 and 54 $<$ heart rate $<$ 100 then this handicap person will benefit from hospitalization service.

   Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:greaterThan(?v, 21) ŝwrlb:lessThan(?vo, 96) ŝwrlb:greaterThan(?vh, 54) ŝwrlb:lessThan(?vh, 100) Ĥospitalize(?ser) -> Benifit_from_hospitalization(?h, ?ser)

7. **Rule 7:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : 11 $<$ respiratory $<$ 22 and oxygen $<$ 96 and 54 $<$ heart rate $<$ 100 then this handicap person will benefit from doctor call service.

   Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh)

Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:greaterThan(?v, 11) ŝwrlb:lessThan(?v, 22) ŝwrlb:lessThan(?vo, 96) ŝwrlb:greaterThan(?vh, 54) ŝwrlb:lessThan(?vh, 100) Ĉall_doctor(?ser) -> Benifit_from_doc_call(?h, ?ser)

8. **Rule 8:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : respiratory < 12 and oxygen < 96 and heart rate < 55 then this handicap person will benefit from hospitalization service.

   Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:lessThan(?v, 12) ŝwrlb:lessThan(?vo, 96) ŝwrlb:lessThan(?vh, 55) Ĥospitalize(?ser) -> Benifit_from_hospitalization(?h, ?ser)

9. **Rule 9:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : respiratory < 12 and oxygen < 96 and heart rate > 100 then this handicap person will benefit from hospitalization service.

   Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:lessThan(?v, 12) ŝwrlb:lessThan(?vo, 96) ŝwrlb:greaterThan(?vh, 100) Ĥospitalize(?ser) -> Benifit_from_hospitalization(?h, ?ser)

10. **Rule 10:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : respiratory > 21 and oxygen > 95 and 54 < heart rate < 100 then this handicap person will benefit from doctor call service.

    Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:greaterThan(?v, 21) ŝwrlb:greaterThan(?vo, 95) ŝwrlb:greaterThan(?vh, 54) ŝwrlb:lessThan(?vh, 100) Ĉall_doctor(?ser) -> Benifit_from_doc_call(?h, ?ser)

11. **Rule 11:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : 22 > respiratory > 11 and oxygen > 95 and heart rate > 150 then this handicap person will benefit from hospitalization service.

    Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh,

?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:greaterThan(?v, 11) ŝwrlb:lessThan(?v, 22) ŝwrlb:greaterThan(?vo, 95) ŝwrlb:greaterThan(?vh, 150) Ĥospitalize(?ser) -> Benifit_from_hospitalization(?h, ?ser)

12. **Rule 12:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : 22 > respiratory < 11 and oxygen < 96 and 54 < heart rate > 150 then this handicap person will benefit from hospitalization service.

    Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:greaterThan(?v, 11) ŝwrlb:lessThan(?v, 22) ŝwrlb:lessThan(?vo, 96) ŝwrlb:greaterThan(?vh, 150) Ĥospitalize(?ser) -> Benifit_from_hospitalization(?h, ?ser)

13. **Rule 13:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : respiratory > 21 and oxygen < 96 and 54 < heart rate < 100 then this handicap person will benefit from hospitalization service.

    Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:greaterThan(?v, 21) ŝwrlb:lessThan(?vo, 96) ŝwrlb:lessThan(?vh, 55) Ĥospitalize(?ser) -> Benifit_from_hospitalization(?h, ?ser)

14. **Rule 14:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : 22 > respiratory > 11 and oxygen > 95 and heart rate < 55 then this handicap person will benefit from food service.

    Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:greaterThan(?v, 11) ŝwrlb:lessThan(?v, 22) ŝwrlb:greaterThan(?vo, 95) ŝwrlb:lessThan(?vh, 55) F̂ood(?ser) -> Benifit_from_food(?h, ?ser)

15. **Rule 15:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : respiratory < 12 and oxygen > 95 and heart rate > 100 then this handicap person will benefit from drug.

    Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:lessThan(?v, 12) ŝwrlb:greaterThan(?vo, 95) ŝwrlb:greaterThan(?vh, 100) T̂ake_drug(?ser) -> Benifit_from_drug(?h, ?ser)

16. **Rule 16:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : respiratory > 21 and oxygen < 96 and heart rate > 100 then this handicap person will benefit from hospitalization service.

    Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:greaterThan(?v, 21) ŝwrlb:lessThan(?vo, 96) ŝwrlb:greaterThan(?vh, 100) Ĥospitalize(?ser) -> Benifit_from_hospitalization(?h, ?ser)

17. **Rule 17:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : 22 > respiratory > 11 and oxygen < 96 and heart rate < 55 then this handicap person will benefit from food and drug.

    Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:greaterThan(?v, 11) ŝwrlb:lessThan(?v, 22) ŝwrlb:lessThan(?vo, 96) ŝwrlb:lessThan(?vh, 55) T̂ake_drug(?ser) F̂ood(?ser2) -> Benifit_from_drug(?h, ?ser) B̂enifit_from_food(?h, ?ser2)

18. **Rule 18:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : respiratory < 12 and oxygen < 96 and heart rate > 150 then this handicap person will benefit from hospitalization service.

    Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:lessThan(?v, 12) ŝwrlb:lessThan(?vo, 96) ŝwrlb:greaterThan(?vh, 150) Ĥospitalize(?ser) -> Benifit_from_hospitalization(?h, ?ser)

19. **Rule 19:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : respiratory > 21 and oxygen > 95 and heart rate < 55 then this handicap person will benefit from drug and food.

    Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:greaterThan(?v, 21) ŝwrlb:greaterThan(?vo, 95) ŝwrlb:lessThan(?vh, 55) T̂ake_drug(?ser) F̂ood(?ser2) -> Benifit_from_drug(?h, ?ser) B̂enifit_from_food(?h, ?ser2)

20. **Rule 20:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : 22 > respiratory > 11 and oxygen < 96 and heart rate > 100 then this handicap person will benefit from drug.

Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:greaterThan(?v, 11) ŝwrlb:lessThan(?v, 22) ŝwrlb:greaterThan(?vo, 95) ŝwrlb:greaterThan(?vh, 100) T̂ake_drug(?ser) -> Benifit_from_drug(?h, ?ser)

21. **Rule 21:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : 22 > respiratory > 11 and oxygen < 96 and heart rate > 100 then this handicap person will benefit from drug.

    Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:greaterThan(?v, 11) ŝwrlb:lessThan(?v, 22) ŝwrlb:lessThan(?vo, 96) ŝwrlb:greaterThan(?vh, 100) T̂ake_drug(?ser) -> Benifit_from_drug(?h, ?ser)

22. **Rule 22:** If an handicap person has 3 sensors connected : respiratory, oxygen and heart rate. and those sensors have measurements : respiratory < 12 and oxygen > 95 and 54 < heart rate < 100 then this handicap person will benefit from drug.

    Handicap_patient(?h) R̂ESP_Sensor(?s) Ôxygene_sensor(?so) ĤR_sensor(?sh) Ĥas_resp_sensor(?h, ?s) ĥas_oxygene_sensor(?h, ?so) ĥas_hr_sensor(?h, ?sh) Ĥas_resp_meas(?s, ?m) Ĥas_oxygene_meas(?so, ?mo) Ĥas_hr_meas(?sh, ?mh) v̂alue(?m, ?v) v̂alue(?mo, ?vo) v̂alue(?mh, ?vh) ŝwrlb:lessThan(?v, 12) ŝwrlb:greaterThan(?vo, 95) ŝwrlb:greaterThan(?vh, 54) ŝwrlb:lessThan(?vh, 100) T̂ake_drug(?ser) -> Benifit_from_drug(?h, ?ser)

3.6 Genetic Algorithm

Due to the high resources consumption and the large amount of information that can be gathered from persons which affect the processing time. We reduced the number of rules without affecting the accuracy of the system. As we discussed in the previous chapter back propagation and descent gradient an be used only in NNs while naive bayes can consume a lot of time due the high number of iteration, This made us choosing the genetic algorithm to decide which rules we will use.

3.6.1 Initialization and chromosome design

The first step is to create and initialize the chromosomes in the population. Since the GA is a stochastic optimization method, the chromosomes's genes are usually initialized at random, so the chromosome is composed only of the feature binary position. In our case the chromosome size is the number of our SWRL rules that we want to select from.

3.6.2 Fitness assignment

After initializing the population, we have to evaluate each chromosome based on a fitness function set by the system designer, And in our case the objective of our system is to reduce the number of rules without loosing the accuracy. The chromosome that gives us a good result comparing to our objective will have a higher rank and the fitness of the chromosome is calculated using the following equation : Where ω is a constant

$$\Phi(i) = \omega * R(i), i = 1 \cdots N$$

called selective pressure, and its value is set between 1 and 2 Higher values of selective pressure will allow the fittest chromosomes to have a higher probability of crossover. The parameter R(i) [Whitley, 1989] is the rank of chromosomes i. The value $\omega = 1.5$ is chosen to calculate the fitness value.

3.6.3 Selection

Once fitness assignment has been finished, the selection operation starts which consisting of choosing two chromosomes that will recombine for the next generation, We keep selecting chromosomes till we reach half of the population. The chromosomes with higher fitness score are most likely to be selected. The selection method that we will use is one of the most used selection methods. It is the roulette wheel [Lipowski & Lipowska, 2012]. This method places all chromosomes on a roulette wheel and turn so the chromosomes are selected randomly, See the Figure bellow

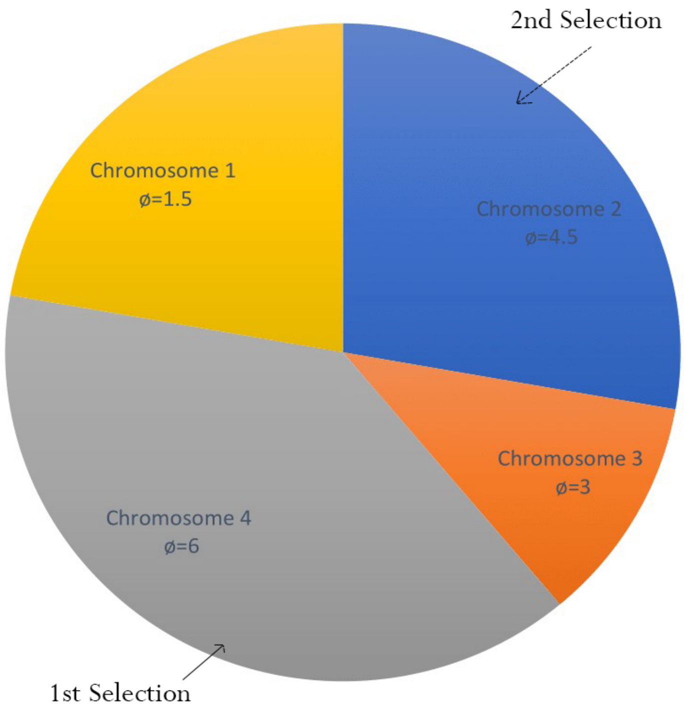


Figure 3.2: Selection Wheel

3.6.4 Crossover

After selecting the chromosomes the crossover operator recombines them to generate new population. We chose Uniform crossover [Spears & De Jong, 1995] which picks two chromosomes randomly and recombine them to obtain four offsprings. This process keeps executing till the new population reaches the same size as the old one.

3.6.5 Mutation

The offsprings resulted from crossover can be very similar to their chromosomes parents which will affect the diversity of results. The mutation process can solve this problem by changing the value of a randomly chosen bit from the offspring. To decide either the mutation will be executed or not we generate a random number between 0 and 1 and if this number is less than the mutation rate that is defined by Janson and Frenzel [Janson & Frenzel, 1993], The bit is flipped. We fixed the mutation rate to 1/m where m is the size of chromosome. [Moghadampour, 2011]

3.7 Conclusion

At the end, in this chapter, we have shown the architecture of our proposed system, starting from data gathering and passing by semantic processing then the genetic algorithm for optimization. We explained concepts, relationships, and rules used for reasoning, and how can the genetic algorithm optimize the number of rules that can preserve resource consumption and processing time. Next, we will discuss the implementation of the system with the tools used for development and results.

Chapter 4

Implementation

4.1 Introduction

After explaining the details of our system conception in the previous chapter, We are going to present The environment of development and the libraries used in our system. Explanation of the main parts of code and results.

4.2 Environment

Hardware

- **CPU :** Ryzen 3 3200g
- **GPU :** Vega 8
- **RAM :** 16 GB
- **Drive :** 256 GB NVMe + 500 HDD

4.3 Development Tools

4.3.1 Protege

Protege is a free and open source ontology editor and a framework for building smart systems. It is a software supported by a large academic community, government and corporate users who use it to build knowledge based solutions in various fields such as healthcare , e-commerce and organizational modeling. You can download Protege for free from http ://protege.stanford.edu. We are using the version 5.5 in our development.

4.3.2 Visual Studio Code

Visual Studio Code is a powerful source code editor which runs on computers and it is available for free download in Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C sharp, Java, Python, PHP, Go) and runtimes (such as .NET and Unity). [Microsoft, 2021]

4.4 Language of programming

4.4.1 Python

Python is an interpreted, object-oriented and high level programming language. It is an easy language to learn due to its syntax simplicity. Python supports modules and packages which encourages program modularity and re-usability of code.It contains a lot of libraries used in machine learning and IA. Developers fall in love with this programming language because of the increased productivity it provides. [https://www.python.org/doc/essays/blurb/]

4.5 Libraries

- **Owlready2 :** It is a package for manipulating OWL 2.0 ontologies in Python. It can load, modify, save ontologies, and it supports reasoning via HermiT and Pellet. Owlready allows a transparent access to OWL ontologies.
- **CSV :** The csv module implements classes to read and write tabular data in CSV format.
- **Names :** Generate random names depending on gender.
- **Math :** This module provides access to the mathematical functions defined by the C standard.
- **Tkinter :** The tkinter library is one of Python's interface packages. tkinter is available on most Unix platforms, including macOS, as well as on Windows systems.

4.6 Data set

BIDMC : This dataset contains vital signals extracted from the much larger MIMIC II matched waveform Database, along with manual breath annotations made from two annotators, using the impedance respiratory signal. [https://physionet.org/content/bidmc/1.0.0/]

4.7 Development of the system

4.7.1 Importing libraries

Before doing anything we should import the libraries we are going to use on the first script which is extractor.py that loads ontology and data gathered from a csv file. Then generate individuals in the ontology based on this data.

```
C: > Users > Ahmed Amin Badji > Desktop > pfe v2 > extractor.py > ...
#LIBRARIES
import names,csv
import owlready2 as owl
import math

print("Libraries imported. \n")

```

Figure 4.1: Importing Libraries

4.8 The ontological model

4.8.1 Importing the ontology

The next figure shows how to import an ontology in Python

```
#Load Ontology
onto = owl.get_ontology("finale-kh.owl").load()
print("Ontology Loaded.")


```

Figure 4.2: Importing Ontology

4.8.2 Initiating the ontology

As we mentioned earlier, We used the software Protege to create our first ontology. In this section We will discuss the creation of concepts, Proprieties.

1. **Concepts** The next figure shows the concepts we created using protege.

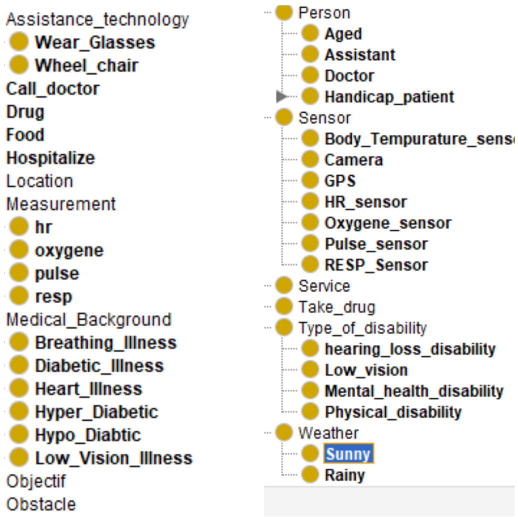


Figure 4.3: Initiating Concepts

2. **Object Proprieties (Relationships)** The next figure shows the relationships we created using protege.

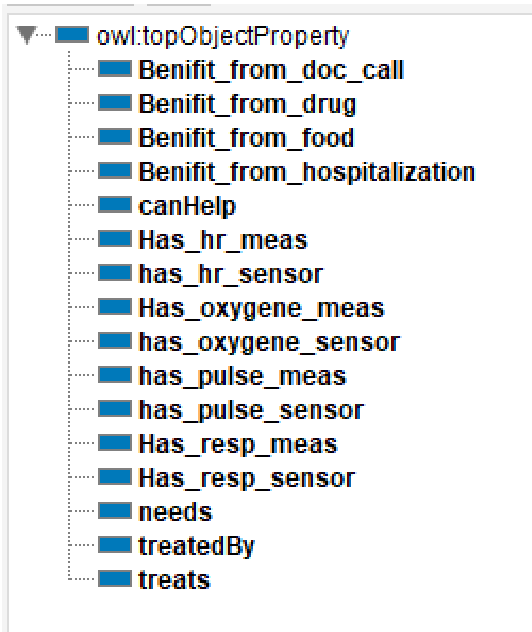


Figure 4.4: Initiating Object Proprieties

3. **Data Proprieties** The next figure shows the data proprieties we created using protege.

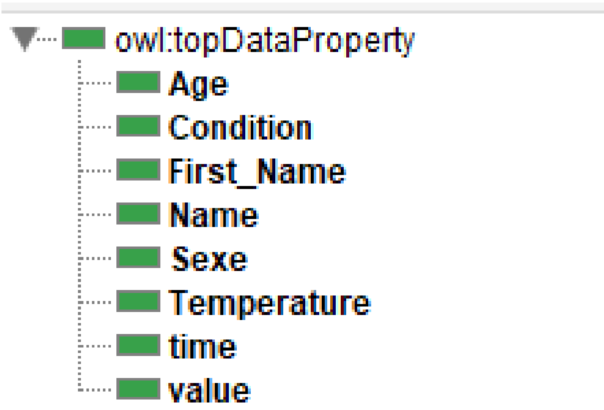
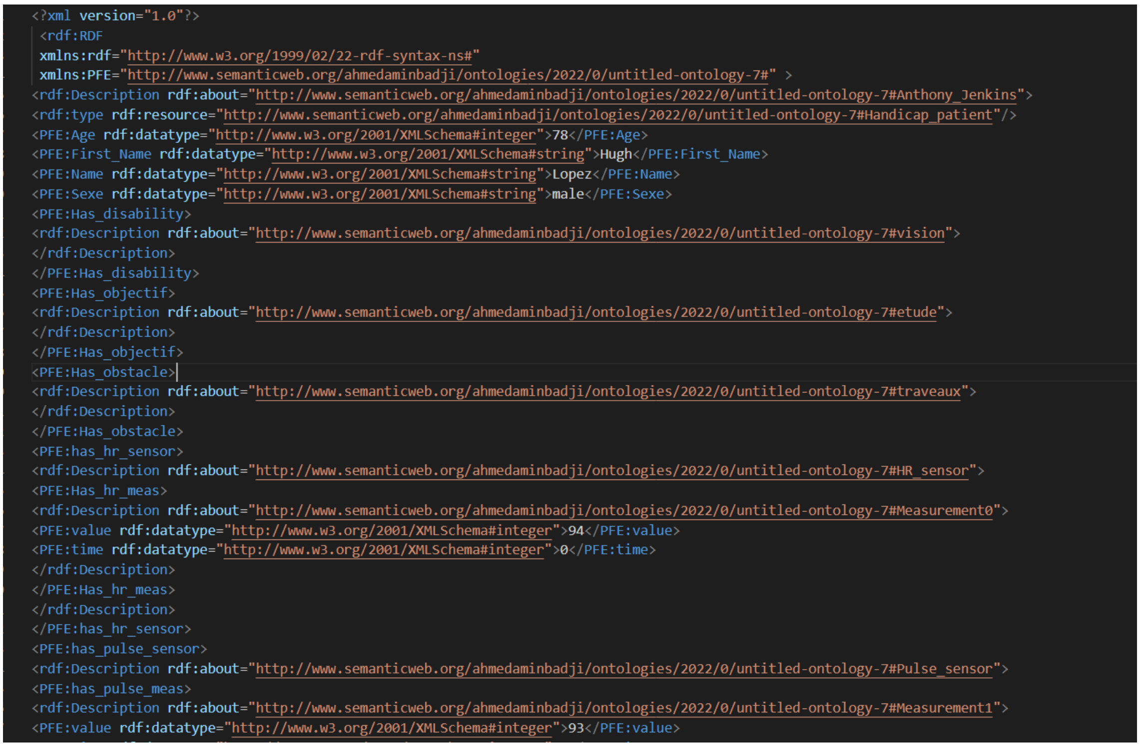


Figure 4.5: Initiating Data Proprieties

4.8.3 Initiating the ontology

4.8.4 Generating RDF Files for handicap instance

We have generated an RDF file for every handicap instance using python. The files can be used in other systems or to visualize the graph related to one handicap. The figure bellow show the structure of a generated file



```
<?xml version="1.0"?>
 <rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:PFE="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#" >
<rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#Anthony_Jenkins">
<rdf:type rdf:resource="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#Handicap_patient"/>
<PFE:Age rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">78</PFE:Age>
<PFE:First_Name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Hugh</PFE:First_Name>
<PFE:Name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Lopez</PFE:Name>
<PFE:Sexe rdf:datatype="http://www.w3.org/2001/XMLSchema#string">male</PFE:Sexe>
<PFE:Has_disability>
<rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#vision">
</rdf:Description>
</PFE:Has_disability>
<PFE:Has_objectif>
<rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#etude">
</rdf:Description>
</PFE:Has_objectif>
<PFE:Has_obstacle>
<rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#traveaux">
</rdf:Description>
</PFE:Has_obstacle>
<PFE:has_hr_sensor>
<rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#HR_sensor">
<PFE:Has_hr_meas>
<rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#Measurement0">
<PFE:value rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">94</PFE:value>
<PFE:time rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">0</PFE:time>
</rdf:Description>
</PFE:Has_hr_meas>
</rdf:Description>
</PFE:has_hr_sensor>
<PFE:has_pulse_sensor>
<rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#Pulse_sensor">
<PFE:has_pulse_meas>
<rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#Measurement1">
<PFE:value rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">93</PFE:value>
```

Figure 4.6: Generated RDF Example

4.9 System overview

In this section we will present the main functions in our system. First of all we need to enrich the ontology with data such as patients, Doctors, And assistants .. etc

4.9.1 Reading information

Due to the lack of data sets related to our work, We decided to use only vital signs of Handicap patients to run our system. In this section We will show you how did we extract the information from BIDMC data set.
The data set contains multiple files:

- Text file containing the age and gender of the person.
- CSV file containing vital signs we will use to enrich our system. The csv file contains about 400 seconds of continuous measurements, But we will be using only the first one. As shown in The figure we are storing the data in an array to use it when we create individuals.

```
filename = "bidmc_csv/bidmc_"+n+"_Numerics.csv"
with open(filename, 'r') as csvfile:
    so = csv.reader(csvfile, delimiter=',', quotechar='"')
    so_data = list(so)
    del so_data[0]
    into.setData(so_data[0])
```

Figure 4.7: Reading Collected Data

4.9.2 Creating Individuals

After reading and storing information we will start creating individuals. There is some difference in programming modules that uses knowledge. For example : In Java we need to work on RDF files for each instance.However, In Python the OWL file is enough to run our system. So we decided to generate both, Individuals in the OWL file and RDF Files for each instance.

- Generating names of handicap patients using gender we saved earlier and creating individual using the ontology object.
- Generating names of Doctors of the patients using the same gender we saved earlier and creating individual using the ontology object.
- Generating names of Assistants using also the same gender we saved earlier and creating individual using the ontology object.
- Adding relationships between the handicap and his doctor and assistant.
- Saving the ontology

```
#GENERATE NAME FROM GENDER
name = names.get_full_name(gender=g)
fname = name.split()[0]
name = name.split()[1]
#Creating Handicap instance
person = onto.Handicap_patient()
person.Name = [name]
person.First_Name = [fname]
person.Age = [int(age)]
person.Sexe = [g]
```

Figure 4.8: Adding individuals to ontology

```
    #ADD RELATIONSHIPS
    owl.AllDifferent([person, doctor,assistant])
    person.needs = [assistant]
    person.treatedBy = [doctor]
```

Figure 4.9: Relationships between the handicap

```
onto.save(file = "finaleok.owl", format = "rdfxml")
```

Figure 4.10: Saving ontology

4.9.3 Main Interface

We developed a simple interface to our system to make the interaction with the system easier. The interface contains a button to load ontology, A button to show the number of rules in it, A button to run Genetic Algorithm, A button to run reasoning without Genetic Algorithm, A button is for reasoning with Genetic Algorithm, And the last button is to load and read RDF file information. In the right of the interface we have input fields and a button to add handicap to the ontology.

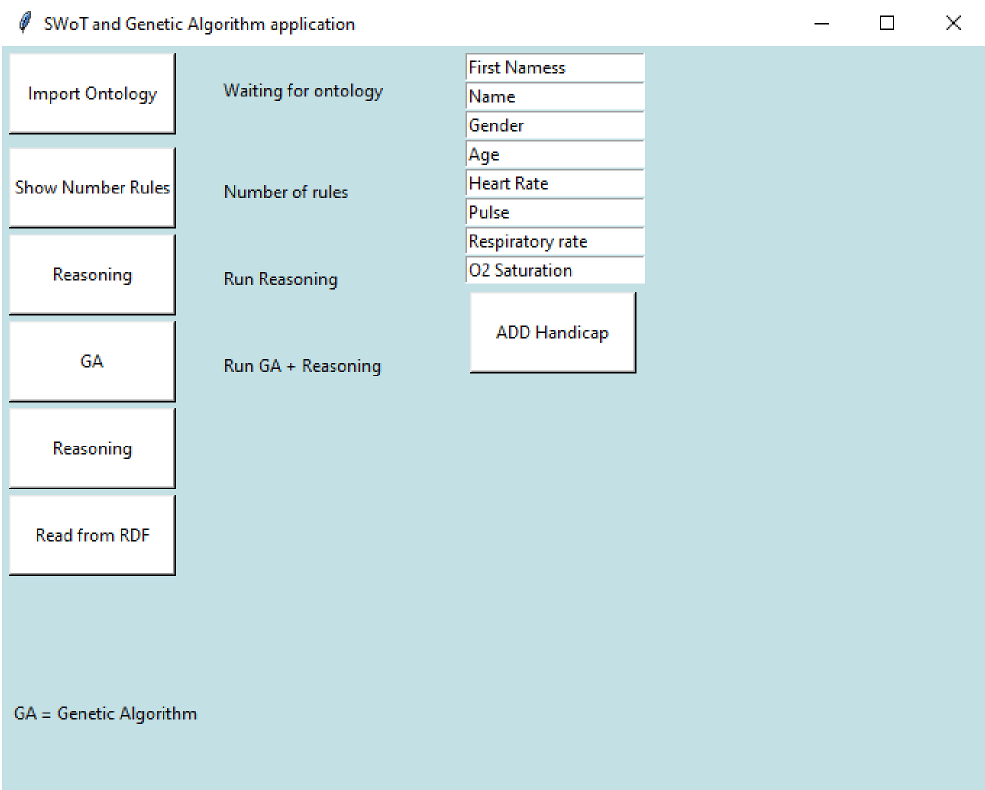


Figure 4.11: Main Interface

4.9.4 Example and results

Before launching the interface of our application, A system initialization is needed. An ontology developed in Protege and a csv dataset. Then we need to run the file extractor.py which will extract data from csv files and add instances to the ontology. After the initialization we can start the application, The main interface will appear. First step to do is loading an ontology using the button Import Ontology. After that you can click on show number of rules so the initial number of rules will be displayed. Everything is setup, The next step is either to add new handicap to the actual ontology using the input fields in the right of the interface then press add handicap button or we can start the Genetic Algorithm to optimize the number of rules. After the optimization process we can run the reasoning without optimization and with optimization and the time of processing will be shown in the interface. After running multiple tests we found that the best test time is reduced by more than 10% with genetic optimization. And about the resource consumption optimization we could not display the exact amount of memory optimized due to the huge amount of codes found in the library of owlready2 and the small community behind it. But after several test we could not execute the reasoning with the default memory limit which is 2000 MB until we changed it to 4000 MB with the full set of rules, However we can execute it with the 2000 MB limit after rules optimization. We will keep this point as a prospective for the upcoming researches.

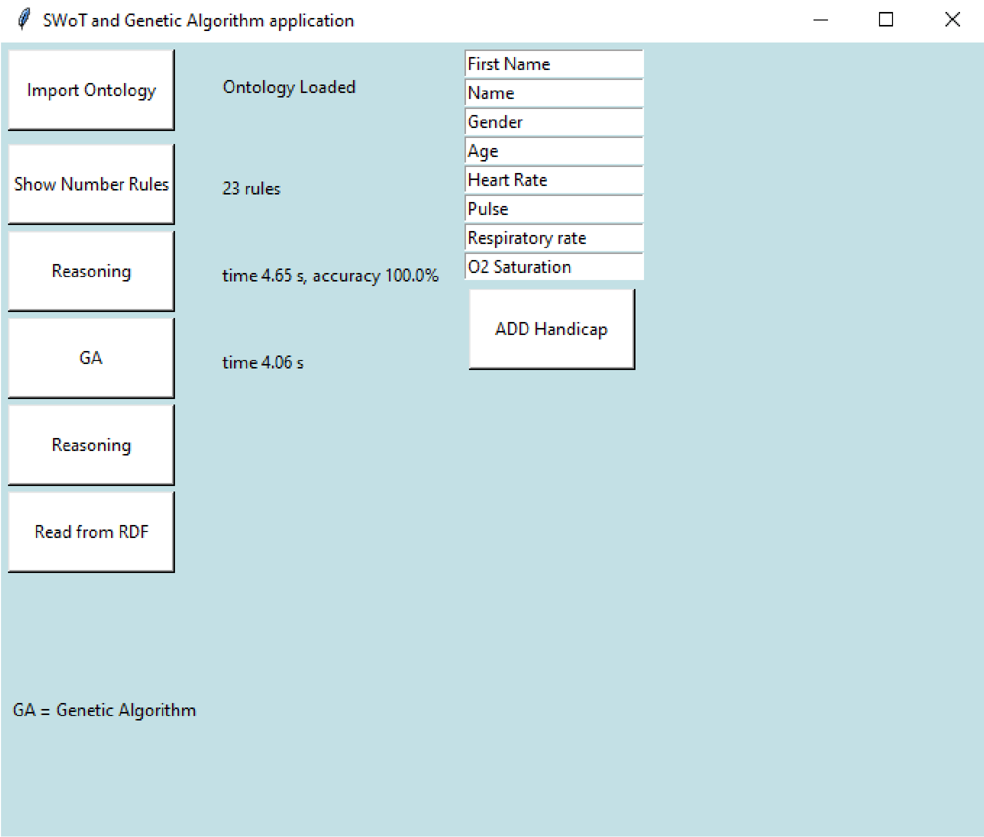


Figure 4.12: Results

General Conclusion

In the goal of solving data heterogeneity problem, We have used the Semantic Web. In addition, A Genetic Algorithm was used to optimize the number of SWRL rules to speed up the reasoning process. We started first by modeling a robust ontology which contains concepts, relationships, and SWRL rules using Protege software. Furthermore, We designed a genetic algorithm that takes N rules and reduce them into less number of rules, which results in reducing the time of semantic reasoning absolutely. Despite the efforts to create this work, our system is still a little bit far from perfection especially because of the lack of resources (Data sets), So we intend in our future works to:

- Testing our system with other data sets.
- Testing our system in a real environment.
- Evaluating our approach in other fields different from medical one.
- Fixing resource consumption problem.

Bibliography

[Akila & Balaganesh, 2021] Akila, D., & Balaganesh, D. 2021. Semantic Web-based critical Healthcare system using Bayesian networks. *Materials Today: Proceedings*.

[Alpes-Maritimes, 2016] Alpes-Maritimes, Département Des. 2016 (03). *Différents types de handicaps*.

[Berners-Lee, 1998] Berners-Lee, Tim. 1998. The Semantic Web road map. 01.

[Burse *et al.*, 2021] Burse, Rashmi, Bertolotto, Michela, O'Sullivan, Dympna, & McArdle, Gavin. 2021. Semantic interoperability: the future of healthcare. *Web Semantics*, 31–53.

[Cambridge Semantics, 2021] Cambridge Semantics. 2021 (07). *Examples of Semantic Web Applications*.

[Casagras, 2009] Casagras. 2009. *CASAGRAS Final Report: RFID and the Inclusive Model for the Internet of Things*. Tech. rept.

[Chatzimichail *et al.*, 2021] Chatzimichail, Angelos, Stathopoulos, Evangelos, Ntioudis, Dimos, Tsanousa, Athina, Rousi, Maria, Mavropoulos, Athanasios, Meditskos, George, Vrochidis, Stefanos, & Kompatsiaris, Ioannis. 2021. Semantic Web and IoT. *Semantic IoT: Theory and Applications*, 3–33.

[Gacitua *et al.*, 2018] Gacitua, Ricardo, Mazon, Jose N., & Cravero, Ania. 2018. Using Semantic Web technologies in the development of data warehouses: A systematic mapping. *WIREs Data Mining and Knowledge Discovery*, **9**(3).

[Goldberg, 1989] Goldberg, David. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. 13th ed. edn. Addison-Wesley Professional.

[Haghi Kashani *et al.*, 2021] Haghi Kashani, Mostafa, Madanipour, Mona, Nikravan, Mohammad, Asghari, Parvaneh, & Mahdipour, Ebrahim. 2021. A systematic review of IoT in healthcare: Applications, techniques, and trends. *Journal of Network and Computer Applications*, **192**, 103164.

[Hogenboom *et al.*, 2009] Hogenboom, Alexander, Milea, Viorel, Frasincar, Flavius, & Kaymak, Uzay. 2009. RCQ-GA: RDF Chain Query Optimization Using Genetic Algorithms. *E-Commerce and Web Technologies*, 181–192.

[i-SCOOP, 2022] i-SCOOP. 2022 (05). *What is IoT? The Internet of Things defined and explained*.

[Jain *et al.*, 2021] Jain, Vishal, Chatterjee, Jyotir Moy, Bansal, Ankita, & Jain, Abha. 2021. *Semantic Web for Effective Healthcare Systems.* Hoboken, NJ, United States: Wiley.

[Janson & Frenzel, 1993] Janson, David J, & Frenzel, James F. 1993. Training product unit neural networks with genetic algorithms. *IEEE Expert*, **8**(5), 26–33.

[Jing, 2016] Jing, Siyuan. 2016 (02). *Architectural evolutionary system based on Genetic Algorithms | Interactive Architecture Lab.*

[Karami & Rahimi, 2019] Karami, Mahtab, & Rahimi, Azin. 2019. Semantic Web Technologies for Sharing Clinical Information in Health Care Systems. *Acta Informatica Medica*, **27**(1), 4.

[Kindberg *et al.*, 2002] Kindberg, Tim, Barton, John, Morgan, Jeff, Becker, Gene, Caswell, Debbie, Debaty, Philippe, Gopal, Gita, Frid, Marcos, Krishnan, Venky, Morris, Howard, Schettino, John, Serra, Bill, & Spasojevic, Mirjana. 2002. People, Places, Things: Web Presence for the Real World. *Mobile Networks and Applications*, **7**(5), 365–376.

[Lapique, 2006] Lapique, Francis. 2006. Le langage d'ontologie web OWL. *Flash Informatique [en ligne]*, 8–24.

[Lipowski & Lipowska, 2012] Lipowski, Adam, & Lipowska, Dorota. 2012. Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications*, **391**(6), 2193–2196.

[Microsoft, 2021] Microsoft. 2021 (11). *Documentation for Visual Studio Code.*

[Moghadampour, 2011] Moghadampour, Ghodrat. 2011. Outperforming mutation operator with random building block operator in genetic algorithms. *Pages 178–192 of: International Conference on Enterprise Information Systems.* Springer.

[Prudhommeaux & Seaborne, 2008] Prudhommeaux, E, & Seaborne, A. 2008. *SPARQL Query Language for RDF.‖ W3C Recommendation 15 January 2008. Retrieved May 24, 2011.*

[Rahmani *et al.*, 2020] Rahmani, Amir Masoud, Babaei, Zahra, & Souri, Alireza. 2020. Event-driven IoT architecture for data analysis of reliable healthcare application using complex event processing. *Cluster Computing*, **24**(2), 1347–1360.

[Roetzel *et al.*, 2020] Roetzel, Wilfried, Luo, Xing, & Chen, Dezhen. 2020. Optimal design of heat exchanger networks. *Design and Operation of Heat Exchangers and their Networks*, 231–317.

[Spears & De Jong, 1995] Spears, William M, & De Jong, Kenneth D. 1995. *On the virtues of parameterized uniform crossover.* Tech. rept. Naval Research Lab Washington DC.

[Takabayashi *et al.*, 2018] Takabayashi, Kento, Tanaka, Hirokazu, & Sakakibara, Katsumi. 2018. Integrated Performance Evaluation of the Smart Body Area Networks Physical Layer for Future Medical and Healthcare IoT. *Sensors*, **19**(1), 30.

[Uschold & Gruninger, 1996] Uschold, Mike, & Gruninger, Michael. 1996. Ontologies: principles, methods and applications. *The Knowledge Engineering Review*, **11**(2), 93–136.

[Wang *et al.*, 2009] Wang, Hua, Ma, Cuiqin, & Zhou, Lijuan. 2009. A Brief Review of Machine Learning and Its Application. *2009 International Conference on Information Engineering and Computer Science.*

[Whitley, 1989] Whitley, Darrell. 1989. The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best. *Pages 116–121 of: Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann.

[World Health Organization, 2020] World Health Organization. 2020 (01). *Disability.*

[Zajmi *et al.*, 2018] Zajmi, Leke, Ahmed, Falah Y. H., & Jaharadak, Adam Amril. 2018. Concepts, Methods, and Performances of Particle Swarm Optimization, Backpropagation, and Neural Networks. *Applied Computational Intelligence and Soft Computing*, **2018**, 1–7.

[Zhong *et al.*, 2015] Zhong, Chang-Le, Zhu, Zhen, & Huang, Ren-Gen. 2015. Study on the IOT Architecture and Gateway Technology. *2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES).*