Demonstration of Semantic Web-based Medical Ontologies and Clinical Decision Support Systems

By

Qiushi Cao

Supervisor

Jan Pettersen Nytun

This master's thesis is carried out as a part of the education at the University of Agder and is therefore approved as a part of this education. However, this does not imply that the University answers for the methods that are used or the conclusions that are drawn.

University of Agder

Faculty of Engineering and Science

Department of Information and Communication Technology

Grimstad, January 5, 2017

Abstract

Ontology represents explicit representation and specification of domain knowledge, and it contains information about objects, entities, concepts and relationships in domains of interest. In this project we design medical ontologies which perform as essential fundamental part of clinical decision support systems, to enable automatic decision support for hospital patients. With the use of semantic web tools and techniques, that are the kernels of this direction, the project aims to develop semantic web-based ontologies and clinical decision support systems, which consist of defined properties, types and interrelationships of a series of objects. These objects include diseases, patients, signs, symptoms and diagnosis. In this work, we address the issue with making diagnosis and classification for patients, according to their clinical signs and symptoms. Knowledge concerning about diseases of spondyloarthritis, inflammatory back pain and diastolic heart failure is represented and modeled.

For disease identifications, we introduce an ontological representation approach to classify patients under these three diseases. The proposed methodology is a combination of both state-of-the-art method which took use of OWL description logic for ontology reasoning, and a novel method which is based on semantic web rule language reasoning and SPARQL rules. With the demonstration of results, we find that all three ontology reasoning approaches are suitable for disease identifications, while semantic web rule language performs more flexible than OWL description logic and SPARQL in terms of maintanance, since Horn-like rules could perform as intermediate data storage entities in the ontology. From results we draw conclusion that knowledge representation in medical domain requires a clear list of medical terminologies and diagnostic criteria as prerequisite, where biomedical terminologies and terms could perform as reasonable sources for developing ontologies. These ontologies aligned with associated generic tools are appropriate for knowledge representation and modeling. For ontology reasoning, all three semantic web technologies are worth considering to implement classification for patients, as they perform promising reasoning functionality.

Keywords: S*emantic web, ontology, knowledge representation, clinical decision support system, ontology reasoning, disease identification*

Preface

This thesis is the result of the Master Thesis course IKT 590 (30 credits), which is part of the ICT Master program at Department of Information and Communication Technology, University of Agder, Norway.

I would like to thank my supervisor associate professor Jan Pettersen Nytun, Ph.D for giving guidance from both technical and writing aspects throughout the project. I would also give thanks to Mukabunani Alphonsine, for her constructive comments and technical support during the thesis work. Without their help, it would be impossible for me to accomplish the main goals of the Master Thesis.

Last but not least, I would like to give sincere gratitude to my parents and friends for their warm-hearted support during the thesis work.

Qiushi Cao

University of Agder

Grimstad, Norway

05 January, 2017

Contents

List of Figures

List of Tables

Abbreviations

DSS	Decision Support System
CDSS	Clinical Decision Support System
CDS	Clinical Decision Support
ICT	Information and Communication Technologies
CPG	Clinical Practice Guideline
OWL DL	Web Ontology Language Description Logic
IBP	Inflammatory Back Pain
DHF	Diastolic Heart Failure
SWRL	Semantic Web Rule Language
SPARQL	SPARQL Protocol and RDF Query Language
AI	Artificial Intelligence
CIG	Computer-interpretable Guidelines
CAD	Computer-assisted diagnosis
RDF	Resource Description Framework
SNOMED CT	Systematized Nomenclature of Medicine-Clinical Terms
ASAS	Assessment of SpondyloArthritis International Society
CHF	Congestive Heart Failure
LVEF	Left Ventricular Ejection Fraction
IRI	Internationalized Resource Identifier

Chapter 1: Introduction

In this chapter, we first present background information about ontology and decision support system (DSS). The background will be unfolded into two parts: concepts and usage. Secondly, motivate factors and problem statements will be described. Thereafter, we demonstrate proposed hypothesis regarding to research questions. At last, we give an outline structure of the master thesis.

1.1 Background and Motivation

Ontology represents explicit representation and specification of domain knowledge, and it contains information about objects, entities, concepts and relationships in domains of interest. Although the original definition of ontology in philosophy is quite theoretical, there exists a set of applications in information and computer science. For instance, ontology is commonly used in knowledge engineering, where ontology in company with a series of individual entities of classes could constitute a knowledge base [1].

Ontologies are embracing a wide usage for precisely describing knowledge bases. Also, ontologies could employ these descriptions in a rich set of applications. These applications are widely used in natural language processing, logic reasoning and decision support systems [2]. Currently, many domains are taking advantages of ontologies, including information and data science, education, environment, biology and medicine. In these fields, life sciences have conspicuously turned into a pioneer of using it, since there are very few scientific domains like life sciences which contain such a huge amount of terms, concepts and definitions. And that, the quantity of terms in these fields are experiencing a significant and rapid growth.

DSS is designed to support decision-making activities in business or organization domains [3]. As being computer-based information systems, decision support systems could assist people and organizations with services in management, operations and maintenance. In reality, decisions could be made by one single authority, or by a group of decision makers [4].

Defined as a health information technology system, a clinical decision support system (CDSS) is designed to improve decision making in clinical domain by providing professionals and physicians with clinical decision support (CDS) [5]. With help of information and communication technologies (ICT), CDSS could generate and store a large-scale knowledge base. Under the environment of hospital wards, for characteristics of individual patients, which are called health observations, CDSS could match decisions with diagnosis and symptoms by the use of a computerized knowledge base [6]. Thus, CDSS is able to deal with these health observations, and then employs software algorithms to manage gathered data, in order to output appropriate clinical decisions as patient-specified recommendations.

While CDSS is composed of powerful software that could offer automatic suggestions on patients thus enhance medical practices, it is not always efficient [6]. There exists limitations in CDSS. For instance, it could be rigescent, and also, when patients have multiple chronic disorders, CDSS does not perform well in the integration of several clinical practice guidelines (CPGs) in terms of management of patients [6]. To solve this problem, ontology was introduced to provide the required flexibility to adapt patient data. Besides, ontologies could provide appropriate decisions presented at a variety of abstraction [6]. The use of ontologies at the core of the system's architecture also proved to enable efficient management of a vast repository of preoperative assessment domain knowledge, including classification of surgical procedures, classification of morbidity and guidelines for routine preoperative tests [7, 8].

Ontologies and CDSSs were first used to develop service models for providing decision support since 1959, and much effort has been made to enhance the performance of these service models. The work we describe in this report focuses on improving and extending the work in [9]. In [9] authors proposed an ontological representation which could specify patient conditions under the disease SpA, and they successfully tested the ontology with the use of web ontology language description logic (OWL DL). Although the preliminary work has proved that OWL DL is a promising technique to carry out ontology reasoning, the research is still at its early stage, and it requires further work.

The following contributions has been made in this thesis work:

- Creating advanced medical ontologies which contain domain knowledge representation of additional diseases like inflammatory back pain (IBP) and diastolic heart failure (DHF).

- Introducing two other novel reasoning approach to make diagnosis and classification for patients, by the use of semantic web rule language (SWRL) and SPARQL Protocol and RDF Query Language (SPARQL)
- Demonstrating the kernels of different ontology reasoning technologies. This work focuses more on what advantages/disadvantages and similarities/differences we could find after assessment of the results.

The advanced ontologies have benefits of containing more medical terminological knowledge, and they enables clients to use three reasoning approaches to make diagnosis for patients. They are also robust for generating other reasoning rules which could be used on extra diseases.

1.2 Problem Statements

The project concerns about the development work of ontologies and CDSS in medicine discipline. It contemplates our efforts in modeling diagnostic criteria through Semantic Web-based approaches, and also the work of demonstration the kernel of these techniques. Semantic Web techniques are used in some ontology based CDSS or representation and reasoning. There are several ways of doing reasoning in Semantic Web, and we focus on using description logic (i.e., OWL DL), rule language (i.e., SWRL) and query/update language (i.e., SPARQL). We investigate existing reasoning approaches and try them out on examples defined by us. Several scientific papers describe reasoning with the use of one of these techniques. A main contribution of this work is to evaluate if our three selected techniques are applicable to the same problem, and by this reveal similarities/differences and advantages/disadvantages. In consideration of this issue, we propose the following research questions:

- *How to create advanced medical ontologies with use of ontology and associated generic tools?*
- *How could we implement ontology reasoning techniques like OWL DL, SWRL and SPARQL, to make diagnosis and disease identification for patients, based on a combination of patients' signs/symptoms?*
- *Compare these three different reasoning approaches, what similarities/differences and advantages/disadvantages could we reveal?*

1.3 Hypotheses

Regarding to research questions proposed in previous subsection, we are going to test the following hypotheses:

- Advanced medical ontologies could be developed through specification and formalization of existing medical terminologies and diagnostic criteria.
- Ontology associated generic tools like Protégé embedded reasoners to check the consistency of the ontology. If we use OWL DL, SWRL and SPARQL as reference technologies, by running these reasoners we could enable automatic diagnosis and disease identification for patients.
- We use Protégé to implement ontology reasoning via SWRL and OWL DL, and Jena Fuseki to execute reasoning through SPARQL. Then we demonstrate analysis based upon the results we get.

1.4 Report Outline

In the remaining thesis, the content is structured as follows:

- Chapter 2 presents underlying concepts and theoretical background related to semantic web technologies and CDSS.
- In chapter 3, we describe the development process of medical ontologies. The development work follows the guideline of knowledge construction and representation process.
- Chapter 4 deals with the reasoning process. We demonstrate three ontology reasoning approaches implemented in this thesis work.
- Chapter 5 gives a comprehensive analysis of the results we get from ontology reasoning. With the demonstration of system outputs, we carry out a discussion upon these results.
- Chapter 6 is devoted to main contributions and conclusions of the thesis.
- In Chapter 7, directions of future research work are proposed.

Chapter 2: Literature Review

In this chapter, we give the review of literatures about CDSSs and ontologies that are used in existing CDSSs. The chapter is composed of three parts. The first subsection is to address the topics about history and development process of CDSSs. In the second subsection we introduce relevant medical terminologies and ontologies. In the last subsection, we demonstrate associated generic tools and standards of ontologies that are used in CDSSs.

2.1 Review of Clinical Decision Support Systems

Nowadays, computer science embraces an extensive participation in medicine and health science. As a branch of computer science, Artificial Intelligence (AI) plays an important role to help medical experts to make clinical decisions [10]. By using the technology of AI, many applications are currently widely used in clinics and hospitals [10]. Although different methodologies and techniques were used during the development process of these systems, any computer program that could offer help to experts and professionals could be considered under the domain of clinical decision support system.

2.1.1 Developmental Stages of Clinical Decision Support Systems

The use of computers to assist medical professionals was executed at the time when computers were invented. The first research paper that concerns about computer and medicine appeared in 1959. In this paper [11], authors analyzed the complicated reasoning processes inherent in medical diagnosis with the aid of electronic computers. Five years after that, an experimental prototype appeared in 1964. During that time period, the limitations of computers' capacity declined the use of computers in medical domain. Later, in 1970s, three advisory systems emerged as original decision systems to provide suggestions: Dombal's system for diagnosis of abdominal pain , Shortliffe's MYCIN system for antibiotics selection and Kuperman's HELP system for medical alerts delivery [10]. After that, an evolution of computer-medicine collaboration embraced a large-scale change from administrative systems to clinical decision support systems [12]. Moreover, in terms of development of architectures for clinical decision support systems, authors in [13] formulated a model with four distinct architectural phases for decision support. Figure 2.1 shows the big picture of evolution of CDSS architectures. The four-phase architecture proposed by authors was sequential and evolutionary, and was proved to be authentic and unambiguous. We can see from Figure 2.1 that the proposed architectural model tracked chronological development of CDSS in a good manner, and it is noted that each phase has gained experience from preceding phases.

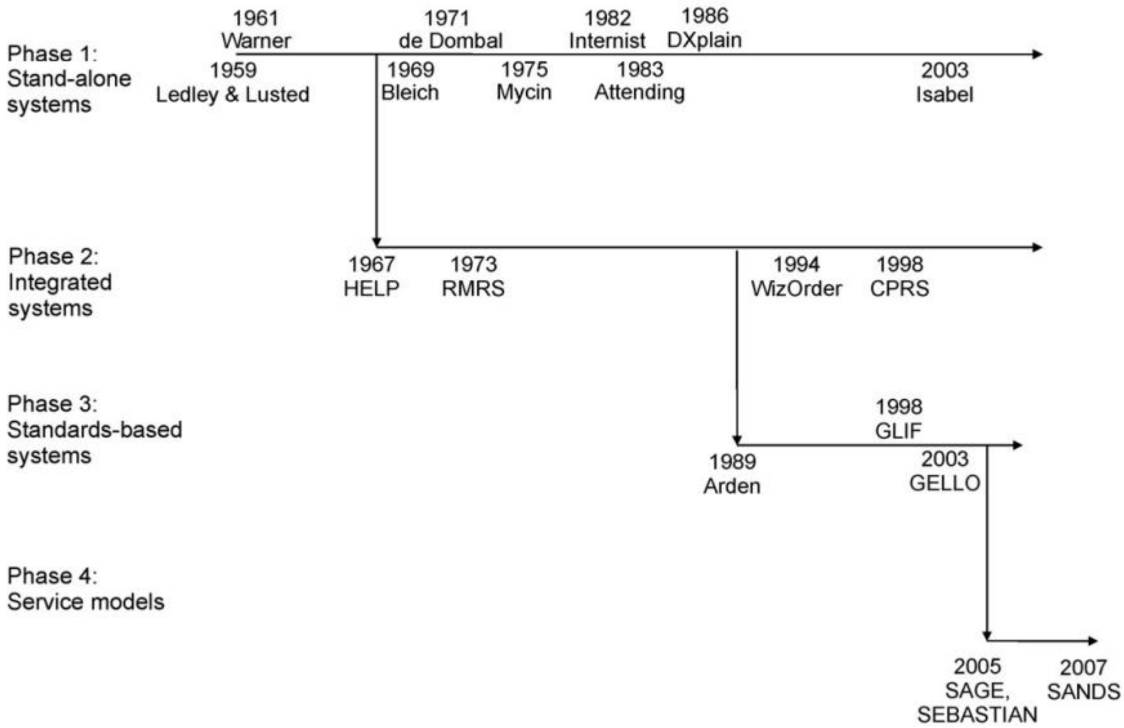

Figure 2.1: A schematic drawing of the four-phase model for clinical decision support [13].

Since the amount of available data from biomedical and knowledge engineering is growing rapidly, there arises the need for sophisticated techniques to cope with this vast quantities of intelligent data. In recent years, with the increasing involvement of computer science in medicine and health sciences, CDSS is more inclined to be designed as a specific class of computerized information systems which could help physicians make decisions and improve quality of healthcare [14, 15]. Many CDSSs have been designed today in an interactive software-based way, and they assisted medical technicians with compiling useful raw data, patients profiles and medical documents. There comes a trend that medical services are becoming more diversified and customized. Embraced the rapid development of technology, clinical and medical services are tailored to provide patients with effective disease prevention and post-treatment management [16].

2.1.2 Existing Use Cases of Clinical Decision Support Systems

Since ICT brings advancement for developing computerized information systems, computer-based CDSSs serve vital important role via promoting clinical convenience and expansion of communication range [16]. With this advantage, a large amount of medical information which is generated and stored in different medical institutions could be maintained and controlled by remote computers. Therefore, healthcare professionals and providers could manage medical patients and cost effectively.

One discussion on different methodologies used in healthcare CDSSs is illustrated in literature [10]. In this paper authors studied the characteristics of CDSS and methodologies through implementation. Figure 2.2 shows representation of different methodological branches of CDSSs.

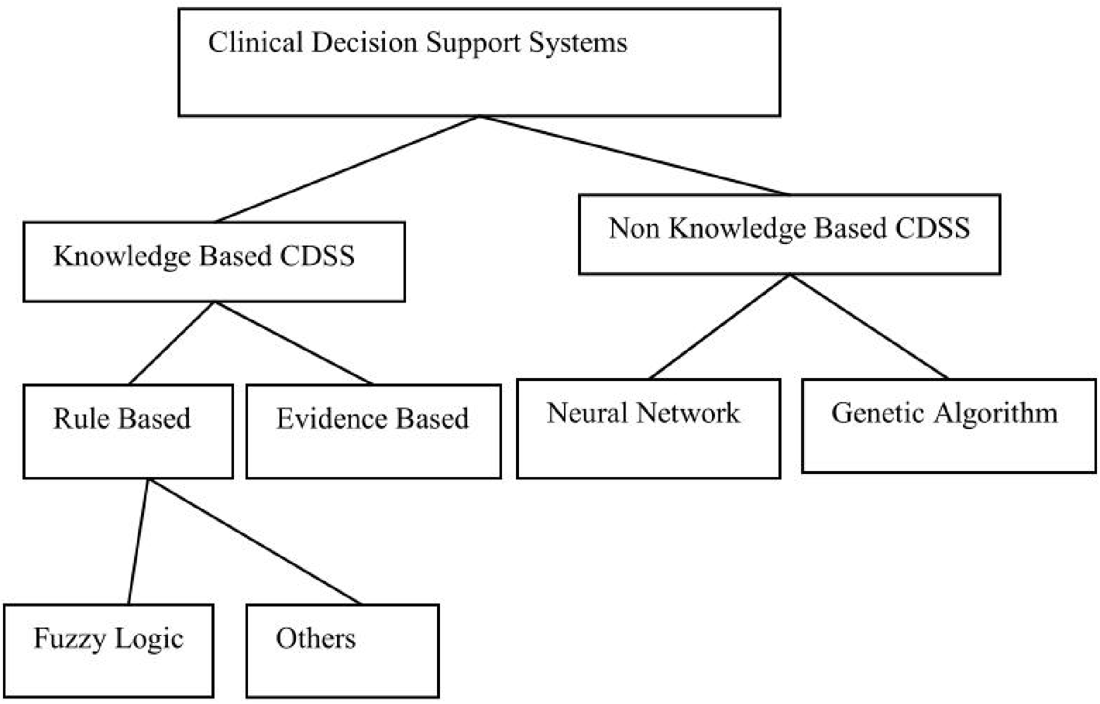

Figure 2.2: Representing different methodological branches of the Clinical Decision Support Systems [16].

From Figure 2.2, authors classified CDSS into two main groups:

- Knowledge based CDSS.
- Non-knowledge based CDSS.

After reading and analyzing 60 research papers, they also concluded that some techniques are domain based, which means these languages and tools are effective only in context of a specific disease area, while others experience an average usefulness in all sorts of domains [16].

While for other researchers' work, Peleg M investigated temporal tendency in the research field of computer-interpretable guidelines (CIGs). Also, he emphasized the important role CIGs played in formalizing CPGs as CIGs, which provided robust functionality than narrative guidelines [17]. With utilization of these techniques, authors made effort in developing an anaphoric relation recognition system. Eadie et al. [18] carried out a survey based on 147 sample research papers related to computer-assisted diagnosis (CAD) with imaging for cancer diagnosis. He discussed variability of sources in terms of the goal of CAD systems, quality of study, learning methodology and variability and so on. Taking advantage of literature analysis, he provided recommendations for researchers, to give assistance with optimizing the quality and comparability of subsequent system design and reporting [18]. Garg et al. [6] concentrated on providing a cumulative summary of controlled trials, and they also evaluated how effective CDSSs could influence medical practitioners' performance and patient outcomes.

2.1.3 Performance and Efficiency of Existing CDSSs

CDSS is defined as a computerized decision-making system, to produce knowledge-based clinical decision support for human resource management [19]. As discussed in subsection 2.1.1, development in computerized information systems has brought advantages in medical services. These advances enable clinical experts and hospital staffs to manage patients' information in a more inexpensive and efficient manner [20]. Also, CDSS could give assistance to clinical experts and hospital staffs to make fast and accurate decisions with avoidance of mistakes [21]. Another research which examined existing workflow patterns, clinical tasks, culture and environment showed improvement of system performance, with implementation of multi-site cross-sectional qualitative study [22]. There is one more study shows the result that one prototype CDSS had an accuracy of 87% while providing recommendations for a random set of patients who represented various decision scenarios [23].

Besides, ICT has a great contribution to user experience of CDSS, by promoting convenience and expansion of scope of communication [21,24,25]. Since there arises the demand of remote monitoring, disease prevention and patient-customized services, specialized medical services combined with ICT provide the possibility of meeting these requirements.

The efforts in publications listed above mostly lie in proposing methods for enhancing performance of CDSSs by either associating or advancing the existing conventional systems. They aimed to develop more accurate and patient-tailored decisions for patients and physicians.

2.2 Terminologies and Ontologies in Biology and Medicine

Medical experts and clinicians have made considerable efforts to address the challenge of dealing with vast amount of biomedical data [26]. To enable automatic reasoning to the large repositories of biomedical knowledge, they have made impressive progress with designing intelligent and efficient information systems to enhance interoperability and expressiveness of medical and biological terminologies/ontologies [26]. As a consequence, there emerges a growing set of semantic reference systems, and they enrich the medical knowledge base in respect to vocabularies, thesauri, terminologies and ontologies [27].

In the domain of medicine and biology, various semantic standards are playing important roles to represent domain knowledge, and biomedical experts have made considerable efforts in describing terms and entities, in order to execute querying and reasoning of intelligent data [26]. These terms and entities aligned to constitute medical and biological terminologies and ontologies.

According to the survey carried out in [26], authors found that SNOMED-CT is a comprehensive and expressive medical terminology, and it is the most popular terminology in medicine domain. Originally designed to cover the whole patient record, SNOMED-CT performs efficient indexing and processing of patient data [28]. As a consequence, domain experts used SNOMED-CT as a standardized controlled terminology, in order to implement translation of medical data. This process is essential for implementation of knowledge-based CDSSs, retrieval of data and aggregation [28].

Officially available in English and Spanish, SNOMED CT is composed of hierarchies of multiple *is-a* atoms, and the whole terminology contains approximately 310,000 nodes [29]. These nodes normally denote concepts in the medicine domain, mostly presented in forms of classes or individuals. Information about diseases, drugs, medical procedures, lab test results are possessed by these medical concepts. Figure 2.3 depicts definition of Cholecystectomy in SNOMED. We can see that from SNOMED CT's criterion, the definition of Cholecystectomy is defined with textual medical terms.

Current Concept:
Fully Specified Name: **Cholecystectomy (procedure)**
ConceptId: **38102005**
Defining Relationships:
Is a **Biliary tract excision (procedure)**
Is a **Operation on gallbladder (procedure)**
Group 1:
Method (attribute): **Excision - action (qualifier value)**
Procedure site - Direct (attribute): **Gallbladder structure (body structure)**
This concept is fully defined.
Qualifiers:
Access (attribute): **Surgical access values (qualifier value)**
Priority (attribute): **Priorities (qualifier value)**
Descriptions (Synonyms):
Preferred: **Cholecystectomy**
Synonyms: **Excision of gallbladder, Gallbladder excision, Removal of gallbladder**
Parents:
Biliary tract excision (procedure)
Operation on gallbladder (procedure)
Children:
Cholecystectomy and exploration of bile duct (procedure)
Cholecystectomy and operative cholangiogram (procedure)
Excision of lesion of gallbladder (procedure)
Laparoscopic cholecystectomy (procedure)
Partial cholecystectomy (procedure)
Total cholecystectomy and excision of surrounding tissue (procedure)

Figure 2.3: SNOMED CT's definition of Cholecystectomy [26].

By the enrichment of ontology representation which is compatible of the OWL DL, domain experts could facilitate the interpretation of these domain knowledge with help of CDSS techniques. In this work, we use medical terms and terminologies like SNOMED CT as knowledge base, then refine and extract knowledge from it, to create our own medical ontologies. These medical ontologies perform as essential fundamental parts of CDSSs.

2.3 Key Technologies in Clinical Decision Support Systems

Since the amount of available data from biomedical and knowledge engineering is growing rapidly, there arises the need for sophisticated techniques to cope with this vast quantities of intelligent data. Biologists, health economists and researchers in biomedicine have made persistent efforts into this challenge. As a consequence, a set of semantic reference systems came onto the stage to perform effective and reliable functionality. These systems are normally characterized by vocabularies, thesauri, terminologies and ontologies [26].

When compared to the past, hodiernal CDSS could perform a more reliable and accurate decision-making motion with the aid of artificial intelligence and other analysis techniques [30]. AI is the key technology used in CDSSs [10]. When this technology is integrated with computer-based CDSS, the new system could be adopted in new environment, and be capable of learning new knowledge as time passes by [31,32].

Authors in [26] summarized the solutions to current developments in biomedical knowledge management, by classifying them into two aspects:

- Establishment of indexing vocabularies and classification systems, which was driven not only by public health and epidemiology interests, also by development of library science.
- The research effort on medical decision support and expert systems, which was driven by the appearance of Artificial Intelligence research work.

They also emphasized the vision of ontology, which has become one of the most fashionable terms in computer science.

2.3.1 Ontology

Domain ontology is normally used to form a knowledge base [33]. Although the original definition of ontology in philosophy is quite theoretical, there exists a set of applications in information and computer science. For instance, ontology is commonly used in knowledge engineering, where ontology in company of a series of individual entities of classes could constitute a knowledge base [34].

Ontologies are embracing a wide usage for precisely and detailedly describing domains. Also, ontologies could employ these descriptions in a rich set of applications. These applications are widely used in natural language processing, logic reasoning and decision support systems [26]. Currently, many domains are taking advantages of ontologies, including information and data science, education, environment, biology and medicine. In these fields, life sciences have conspicuously turned into a pioneer of using it, since there are very few scientific domains like life sciences which contain such a huge amount of terms, concepts and definitions. And that, the quantity of terms in these fields are experiencing a significant and rapid growth.

2.3.2 Semantic Web

"The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.'' [34]

According to Tim Berners Lee and Fischetti, the idea of Semantic Web is to create a layer on the existing web structure, so as to enable advanced automatic processing of data and web content [35]. In this way, structured data is generated and shared with humans and computers, because they are interlinked. Since shared information could be read automatically by computers, they could be connected and queried.

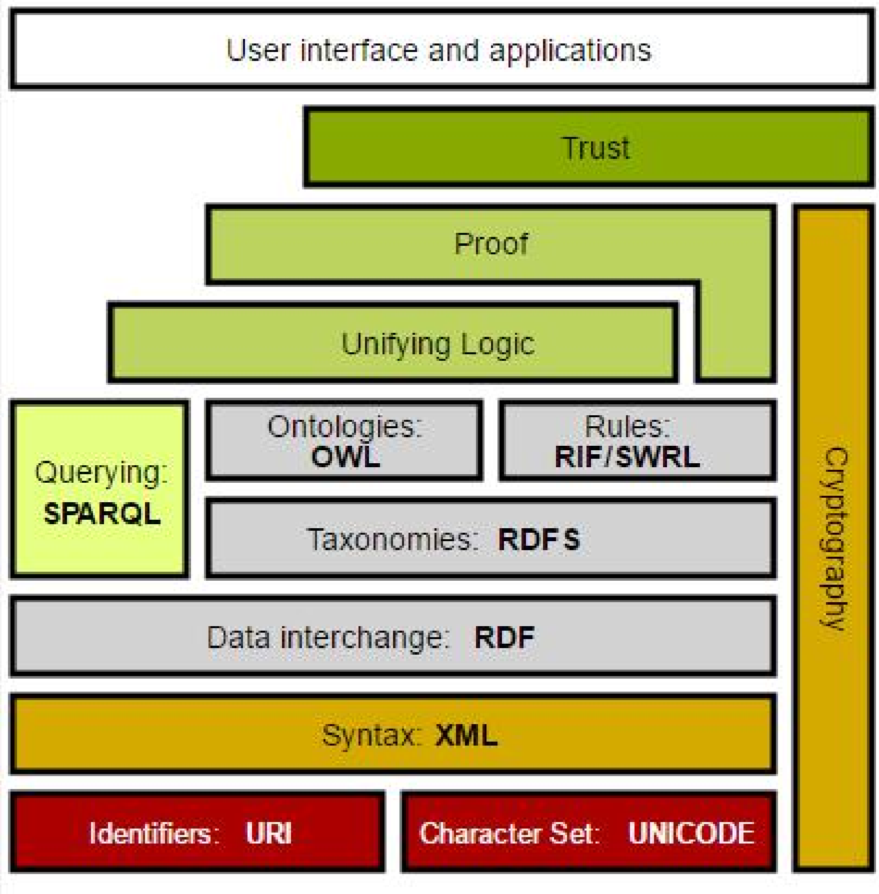

Figure 2.4: The Semantic Web Stack [36].

Figure 2.4 shows the Semantic Web Stack. From this architectural framework, we could observe that different formats and standards are combined together to enable Semantic Web technology. Among these, the most important technologies are Web Ontology Language (OWL), Resource Description Framework (RDF), Semantic Web Rule Language (SWRL) and SPARQL.

2.3.3 Web Ontology Language

Ontology could be formalized in OWL, and OWL is based on a subset of DLs [37]. By providing a wealthy set of vocabularies together with formal semantics, OWL could offer experts great machine interpretability of web content.

OWL is built upon RDF, which is a W3C standard for objects. Since the time it was proposed, both OWL and RDF have attracted great interests in terms of academy, medicine and commerce [38]. The latest version of OWL was announced by W3C in the year 2009, which is called OWL 2. Afterwards, various semantic editors came onto the stage, such as Protégé [39]. Along with them, several semantic reasoners such as Pellet, RacerPro and FaCT++ enabled the inference of logical consequences from a set of asserted facts or axioms [40].

2.3.4 Resource Description Framework

RDF is a directed, labeled graph data format for representing information in the Web [41]. Instead of structuring the syntax of data, this metadata approach from W3C defines semantic meaning for data on the web [42]. To represent distributed data, a triple is a common simple way to show the relationship of two individuals in the web.

Figure 2.5 from [43] illustrates a simple example of a triple. The purple arrow between two named entities denotes the property of ``has individual''.

Figure 2.5: An example of a triple in RDF [43].

2.3.5 Semantic Web Rule Language

SWRL is used as the rule language in semantic web, and it is designed based on a combination of OWL DL and OWL Lite [44]. High-level Horn-like rules could be created via SWRL, and the rules could be written in abstract syntax, in order to implement reasoning on individuals in existing ontologies.

Generally, one rule consists of a premise and a conclusion, which means in any situation if the premise applies, then the conclusion holds. Applying inductive reasoning, SWRL has been proved to be a powerful and promising technique for ontology reasoning, and it also shows some capabilities of expressing statements that can not be written in OWL [45]. With regard to this robust functionality of

SWRL, we decide to choose SWRL as one reasoning approach. One example of rule in our CDSS is as follows:

Patient(?x) ^ presents(?x, Psoriasis) -> has_diagnosis(?x, Family_history_for_SpA).

In this way, we formulate one production rule with the knowledge from medical terminology SNOMED CT, and this rule is meant to be used to reason on instances in our CDSS. The meaning of this rule is: if one P*atient* presents the symptom *Psoriasis,* then the CDSS has *diagnosis* for this *patient,* as *Family_history_for_SpA.* Through creating production rules, we implement diagnostic reasoning on medical ontologies, thus output automatic diagnosis and disease identification for patients.

2.3.6 SPARQL

Short for SPARQL Protocol and RDF Query Language, SPARQL provides a standard way to access RDF data. SPARQL could be used to propose expressive queries to give decision support, no matter whether the data is stored natively as RDF or viewed as RDF via middleware [39]. Figure 2.6 gives an example of a query graph in SPARQL. We can express the graph in SPARQL language as followings format [41]：

Data: *: Jone Snelson :speaking-at "SemTechBiz".*

Query: *Select * where {*

?who: speaking-at "SemTechBiz".

}

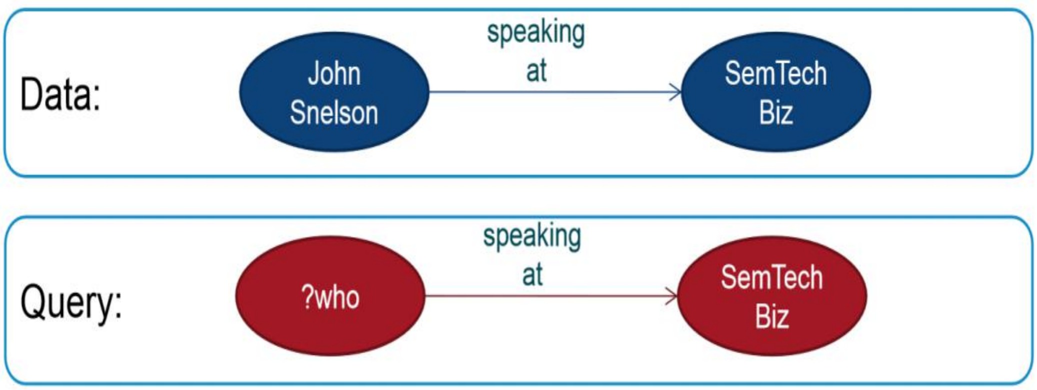

Fig 2.6. An example of SPARQL Query [41].

Through the use of queries which are applied on ontologies, we could know how accurate and expressive the ontology is answering questions from users. In this project, we use SPARQL Rules and Query as tools to classify patients, according to their diagnosis and symptoms.

2.4 Summary of the Chapter

In this chapter, we introduced the literature review in three aspects: CDSS, medical terminologies/ontologies and Semantic Web techniques. With awareness of the fact that there is no medical ontology deals with how to build relationships between signs/symptoms and diagnosis in diseases IBP and DHF, we decided to use three reasoning approaches to cope with this challenge. Also, we plan to extend the patients classification work in [37], in which OWL DL reasoning was introduced to identify the disease SpA. We will further demonstrate our efforts of designing and analyzing the performance of medical ontologies and CDSSs in the next chapter.

Chapter 3: Ontology Development

In this chapter, we are going to explain the development process of ontologies. We decided to follow the instructional knowledge construction and representation process which is introduced in [46], since it has been proved to be efficient and effective. We will firstly show an overview of our ontology model architecture, and then introduce the materials and sources we used in developing the ontology. The sources and materials include medical terminologies and ontologies of interests, and also compass tools and languages we used. Thereafter, a detailed step-by-step design process will be presented.

3.1 Scope and Sources

To start with constructing ontologies, we should first specify conceptualized ideas and desired area of reasoning. Since the starting objective of this thesis work is to construct ontology models which could be used to make diagnosis and disease identification for patients, it requires conceptualization and codification of the domain knowledge, and this domain knowledge should be rigorously recognized by domain professionals. In view of this, Figure 3.1 shows a big picture of ontology development.

As depicted in Figure 3.1, ontology specification process contains methodology of conceptualization. After outlining structure of the ontology, we need to define classes, individuals, object properties and data properties in detail. OWL language was selected as the basic tool in this development work, as it could provide a rich set of knowledge representation functions, such as classes, individuals and properties. Also, OWL has favorable properties like expressive and strongly-readable [46].

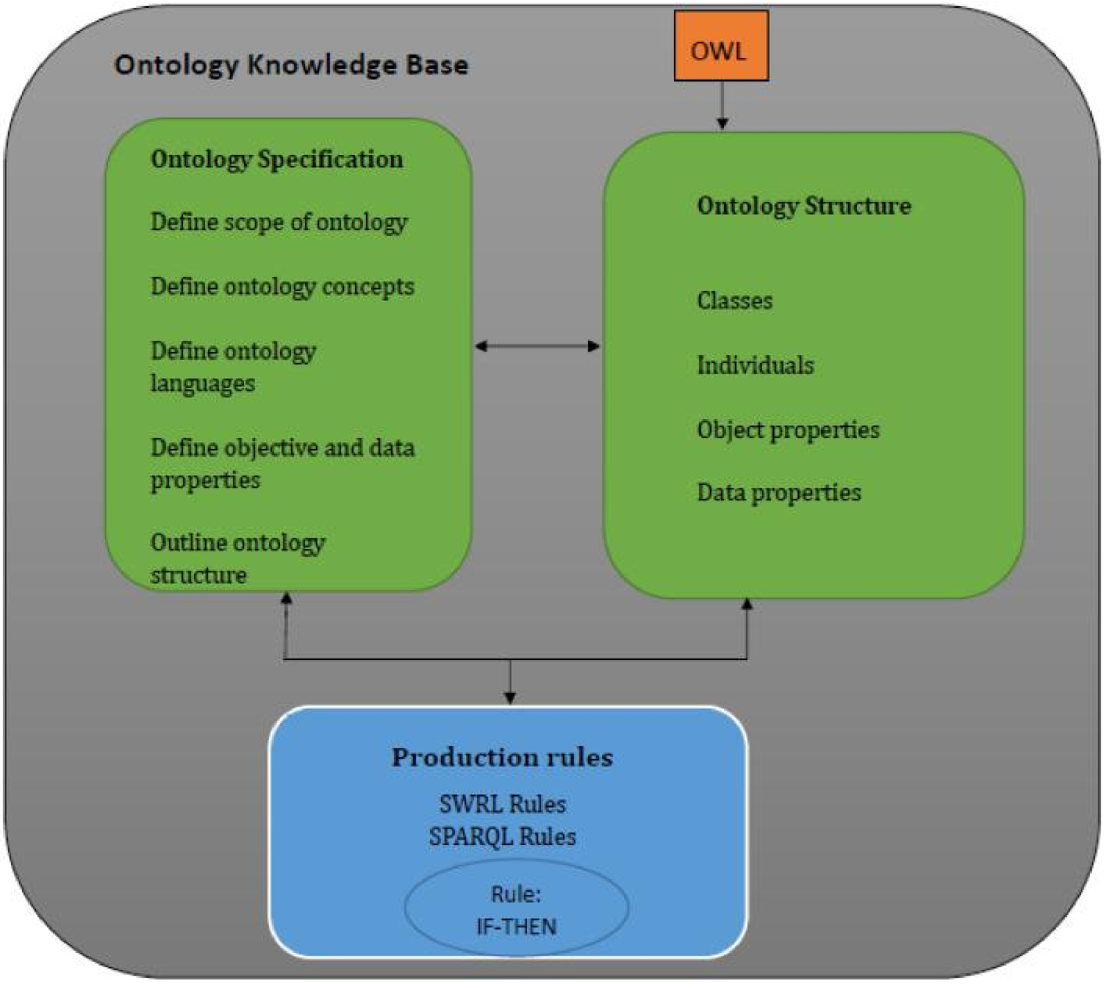

Figure 3.1: Big picture of ontology development.

Normally, there are two approaches to define the scope and structure of ontology: bottom-up approach and top-down approach. Researchers have investigated a lot on both of approaches, and they found bottom-up approach, which starts from patients' records, consumes heavy workload [47]. On contrast, top-down approach has been proved to enable developers to gather concepts and information in an easier way [3]. As a consequence, we decide to choose top-down approach in our work.

When the ontology has been created, production rules in forms of SWRL rules and SPARQL rules are created to enable reasoning of the ontology. The proposed rules are in a form of an implication between antecedent (body) and consequent (head), and the format of these rules are based on IF-THEN logic. These reasoning approaches have been proved to be capable of using for procedural knowledge description, and interoperability enhancement of the ontology [47]. At the same time, suggestions and input from domain experts should consistently be added to the system. Figure 3.2 illustrates the architecture we described above.

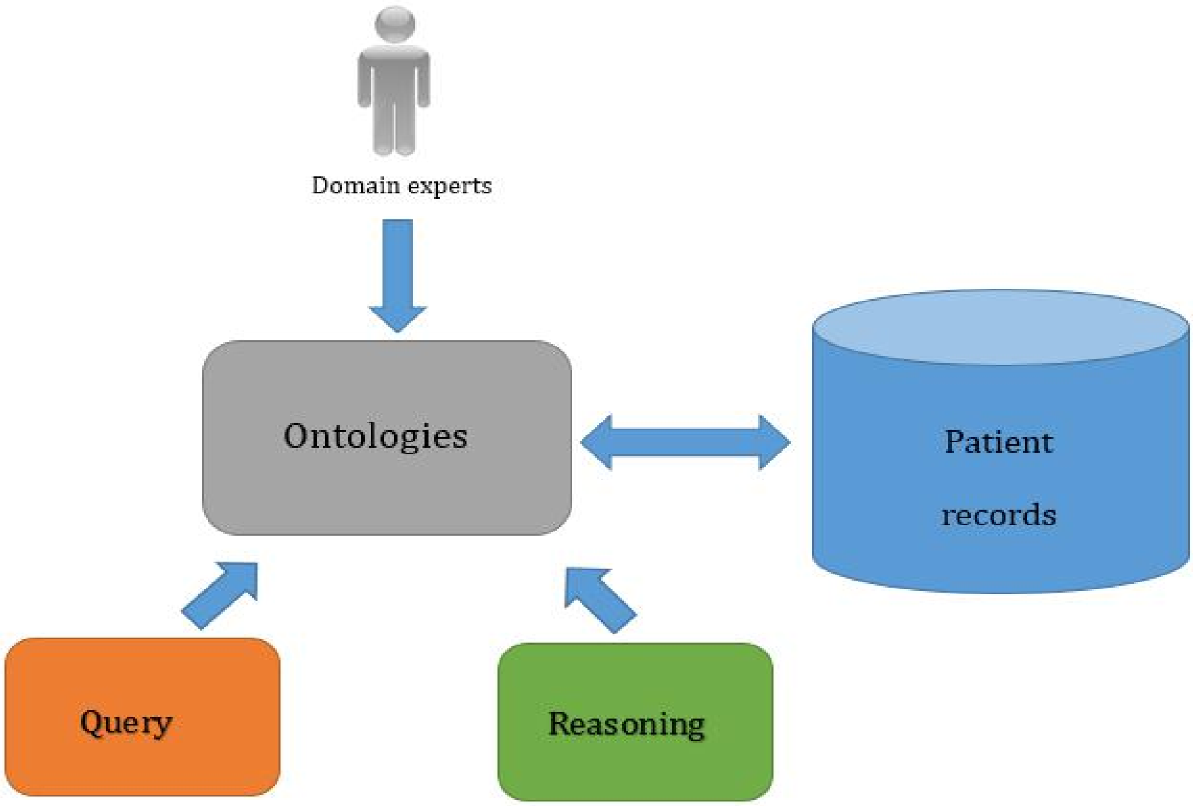

Figure 3.2: Architecture of the designed CDSS.

In the domain of healthcare, we could obtain patients' data from hospital wards, in forms of signs and symptoms, in order to extract interesting information from their history data. These data could be used by domain experts to find relationships between symptoms/signs and diseases, thus provide appropriate treatment and health-care for patients. Also, we could make use of these patient records to extract information and then import it as input to the CDSS, to enable decision making.

After explicating the scope of ontology, the next step is to determine and select appropriate sources of domain knowledge. By the vision of ICT, we would like to take advantage of the abundant data generated by certain diseases. Through knowledge modeling and reasoning, we could reuse others' knowledge to provide guidelines, thus enhance output of ontologies which we are going to develop. As mentioned in Chapter 2, with capabilities of representing formal medical knowledge, medical ontologies and associated generic tools are suitable for achieving the goal of this thesis. Therefore, finding out appropriate medical terminologies or ontologies, which could provide formal representation of diseases, is vital important.

Since we choose SpA, IBP and DHF as diseases of representation in this study, we also selecte Assessment of SpondyloArthritis International Society (ASAS), which is the most used criteria for disease SpA, as one medical terminology.

In ASAS criteria, one patient is diagnosed as potential patient if he or she coincides with two preconditions: suffers from back pain for at least 3 months, and has age at onset less than 45 years old. After qualifying these two preconditions, there are two approaches to classify this patient, one is from Sacroiliitis on imaging and the other is HLA-B27. Figure 3.3 shows how the classification process of SpA in ASAS criteria works.

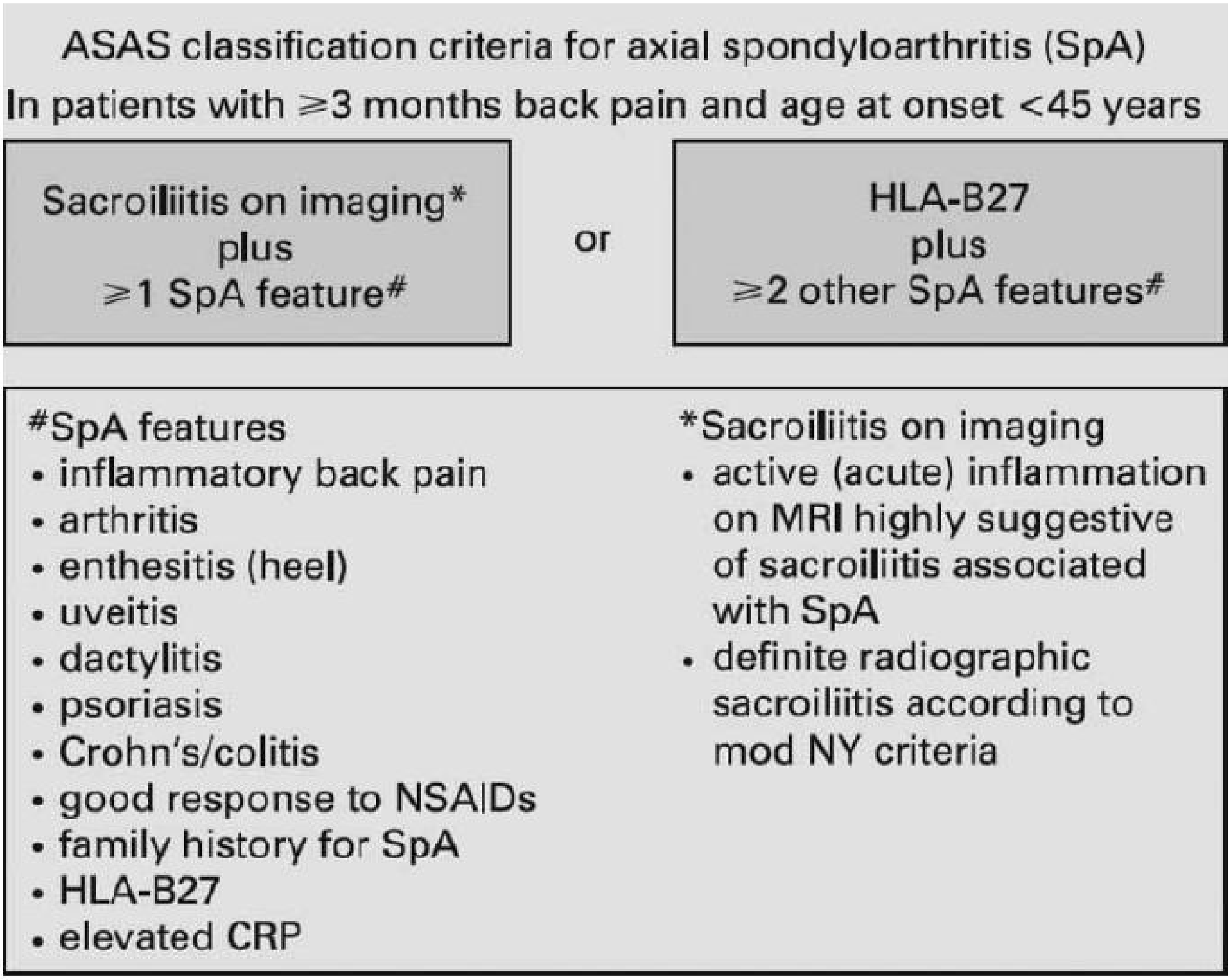

Figure 3.3: ASAS criteria for classification of axial spondyloarthritis [37].

For SpA features, ASAS criteria also provides solutions to specify them. Table 3.1 describes how SpA features and disease IBPcould be defined by signs and symptoms of patients. We can see from Table 3.1 that signs and symptoms are connected with doctors' diagnosis, and this approach could provide assistance for us to translate medical terminologies to ontologies, and then execute knowledge representation based on the abstraction of domain knowledge.

IBP	IBP according to experts (see also Box B): four out of five of the following parameters present: (1) age at onset <40 years, (2) insidious onset, (3) improvement with exercise, (4) no improvement with rest (5) pain at night (with improvement upon getting up)
Arthritis	Past or present active synovitis diagnosed by a doctor
Family history	Presence in first-degree or second-degree relatives of any of the following: (a) ankylosing spondylitis, (b) psoriasis, (c) uveitis, (d) reactive arthritis, (e) inflammatory bowel disease
Psoriasis	Past or present psoriasis diagnosed by a doctor
Inflammatory bowel disease	Past or present Crohn's disease or ulcerative colitis diagnosed by a doctor
Dactylitis	Past or present dactylitis diagnosed by a doctor
Enthesitis	Heel enthesitis: past or present spontaneous pain or tenderness at examination at the site of the insertion of the Achilles tendon or plantar fascia at the calcaneus
Uveitis anterior	Past or present uveitis anterior, confirmed by an ophthalmologist
Good response to NSAIDs	At 24–48 h after a full dose of NSAID the back pain is not present anymore or much better
HLA-B27	Positive testing according to standard laboratory techniques
Elevated CRP	CRP above upper normal limit in the presence of back pain, after exclusion of other causes for elevated CRP concentration
Sacroiliitis by X rays	Bilateral grade 2–4 or unilateral grade 3–4. according to the modified New York criteria (see Part C. Box 2)
Sacroiliitis by MRI	Active inflammatory lesions of sacroiliac joints with definite bone marrow oedema/osteitis suggestive of sacroiliitis associated with spondyloarthritis (see also Part B on MRI)

Table 3.1: Specification of the variables used for the ASAS criteria for classification of axial spondyloarthritis [37].

The third disease of interest is DHF. DHF is defined as a sub-condition of congestive heart failure (CHF), and they both are major public health problems in developed countries. We found an interesting classification schema for DHF proposed in [48]:

- Definite DHF
- Probable DHF
- Possible DHF.

The classification schema was proposed for diagnosis of DHF, according to patients' degree of diagnostic certainty. The classification approach is introduced in Table 3.2- Table 3.4.

Criteria for Definite DHF

Criterion	Objective Evidence
Definitive evidence of CHF AND	Includes clinical symptoms and signs, supporting laboratory tests (such as chest X-ray), and a typical clinical response to treatment with diuretics, with or without documentation of elevated LV filling pressure (at rest, on exercise, or in response to a volume load) or a low cardiac index
Objective evidence of normal LV systolic function in proximity to the CHF event AND	LV EF ≥0.50 within 72 h of CHF event
Objective evidence of LV diastolic dysfunction	Abnormal LV relaxation/filling/distensibility indices on cardiac catheterization

Table 3.2: Criteria for Definite DHF [48].

Criteria for Probable DHF

Criterion	Objective Evidence
Definitive evidence of CHF AND	Includes clinical symptoms and signs, supporting laboratory tests (such as chest X-ray), and a typical clinical response to treatment with diuretics, with or without documentation of elevated LV filling pressure (at rest, on exercise, or in response to a volume load) or a low cardiac index
Objective evidence of normal LV systolic function in proximity to the CHF event BUT	LV EF ≥0.50 within 72 h of CHF event
Objective evidence of LV diastolic dysfunction is lacking	No conclusive information on LV diastolic function

Table 3.3: Criteria for Probable DHF [48].

Criteria for Possible DHF

Criterion	Objective Evidence
Definitive evidence of CHF AND	Includes clinical symptoms and signs, supporting laboratory tests (such as chest X-ray), and a typical clinical response to treatment with diuretics, with or without documentation of elevated LV filling pressure (at rest, on exercise, or in response to a volume load) or a low cardiac index
Objective evidence of normal LV systolic function, but not at the time of the CHF event AND	LV EF ≥0.50
Objective evidence of LV diastolic dysfunction is lacking	No conclusive information on LV diastolic function

Table 3.4: Criteria for Possible DHF [48].

To output diagnosis and disease identification for patients, we specify and conceptualize this medical domain knowledge, and then formalize our ontologies through domain ontology development methods. We used refined and extracted knowledge to propose an ontology-based framework, in order to facilitate clinical decision support.

When the scope and sources specification work is completed, we then could enumerate step-by-step workflow as follows, with the help of the general guideline introduced in [49]：

- Identification of knowledge: determine what domain knowledge is important, and what kinds of knowledge are less important.
- Determination of scope of knowledge: decide what domain knowledge is to be extracted, refined, captured, and formalized.
- Formalization and integration of knowledge: after the scope of knowledge is determined, we could formalize and integrate the knowledge from experts to repositories.
- Representation and storage of knowledge: we could represent and store the domain knowledge to ensure that it is easy to access, expedient to navigate, simple to understand, maintain and reuse.
- Domain knowledge update: when there is new knowledge acquired by users, the domain knowledge of CDSS should be easy to update.

3.2 Tools and Languages

In Chapter 2, we have discussed most popular tools and techniques in semantic web. In this work, we decide to choose Protégé as the tool for creating, modeling and editing ontologies. Since the current version Protégé 5.0 is open-source, and it provides full support of the OWL-2 languages, it enables a rich set of plug-ins like DL-Learner, SWRLTab, Ontograph and so on. Also, reasoners like Pellet, Hermit and FaCT++ perform as inference engines, enabling consistency check of ontologies. Moreover, the user interface is friendly and easy to use.

Protégé also has built-in query tabs, we plane to take use of these query tabs to enable query of ontologies, to achieve the goal of navigation and information retrieval. SPARQL could give us solution of accessing the knowledge from RDF knowledge bases, and we could explore data in a form of queries, to mine unknown relations. In this work, we plan to use SPARQL to retrieve patients' information like object and data properties, and these results will be shown in Chapter 5.

3.3 Ontology Design

In previous subsection, we have selected appropriate tools and languages for the ontology. After that, we could design our ontology via top-down approach.

3.3.1 Classes and Class Hierarchy

Since we follow a top-down class design approach, we introduce higher-level classes first. There are four high-level classes (or super classes)in the ontology: *Patient, Disease, Signs_and_symptoms* and *Diagnosis*. Under super class *Signs_and_symptoms*, there are medium-level classes. For example, for the definition of disease Spondyloarthritis according to ASAS criteria, patient could be diagnosed as having feature of *Sacroiliitis_on_imaging*. And under subclass *Sacroiliitis_on_imaging*, there are two bottom-level classes *Active_inflammation_on_MRI* and *Definite_radiographic_sacroiliitis*. This structure demonstrates that one patient could be classified as having diagnosis *Sacroiliitis_on_imaging* according to two categories. Figure 3.4 displays class hierarchy of the ontology.

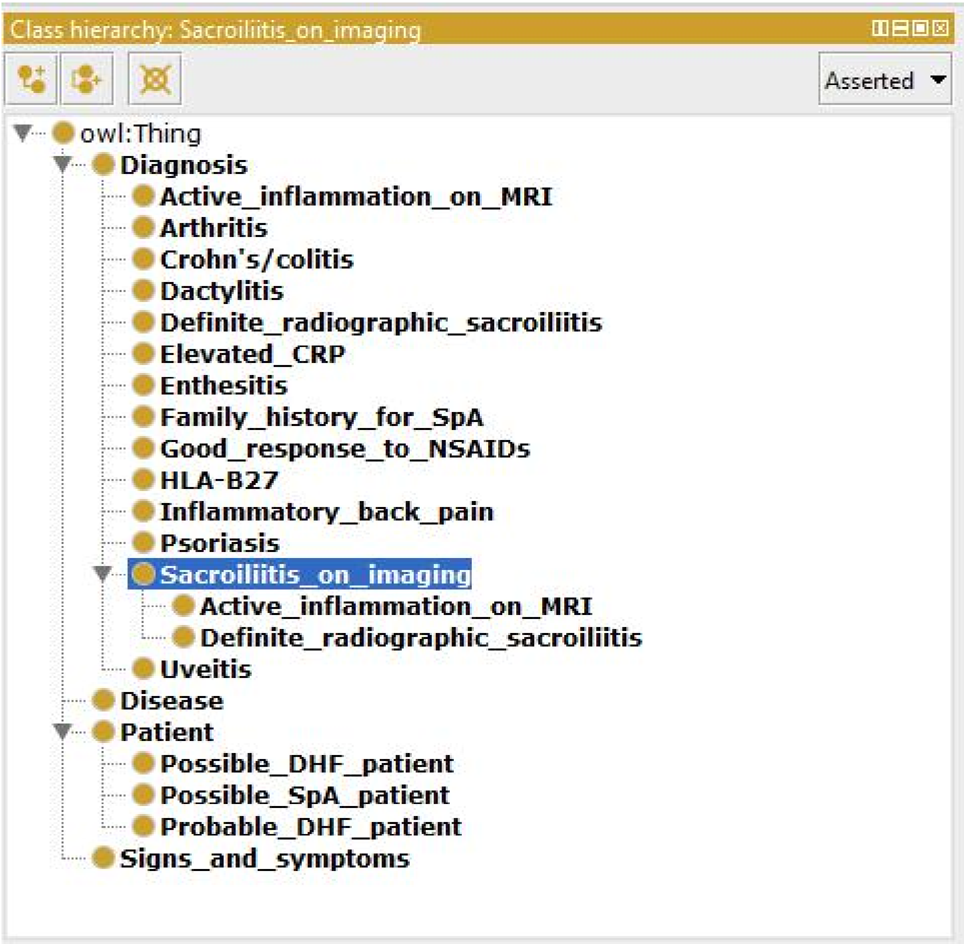

Figure 3.4: Class hierarchy of the designed ontology.

Through the top-down approach, in total 23 classes were created. They align to represent abstract concepts from medical terms and terminologies.

3.3.2 Individuals

Classes could contain individual objects which are called individuals. In our ontology, we create all Signs and symptoms, Patients, Diagnosis and Diseases as individuals. Figure 3.5 and 3.6 depicts some example individuals under classes *Signs_and_symptoms* and *Diagnosis* respectively.

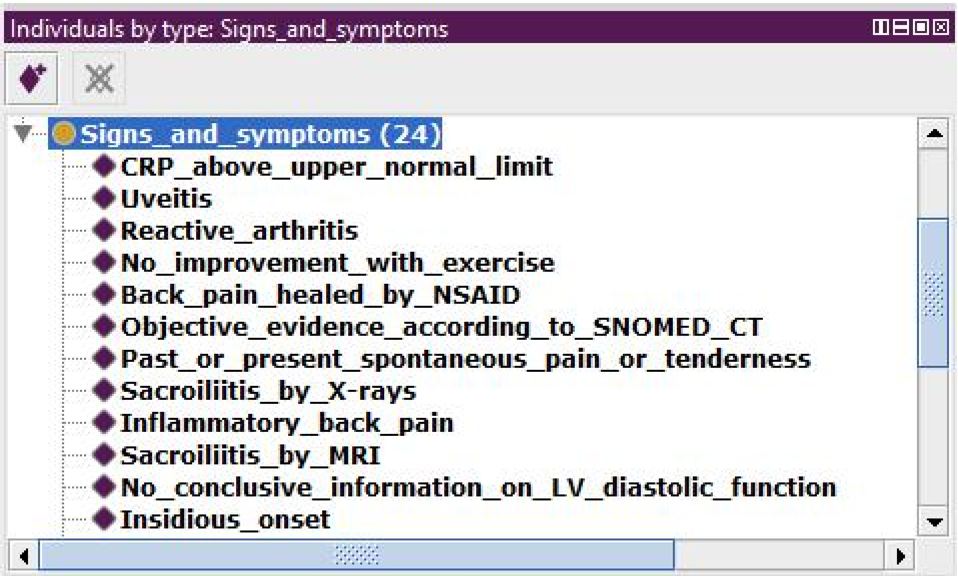

Figure 3.5: Some example individuals under class "Signs_and_symptoms"

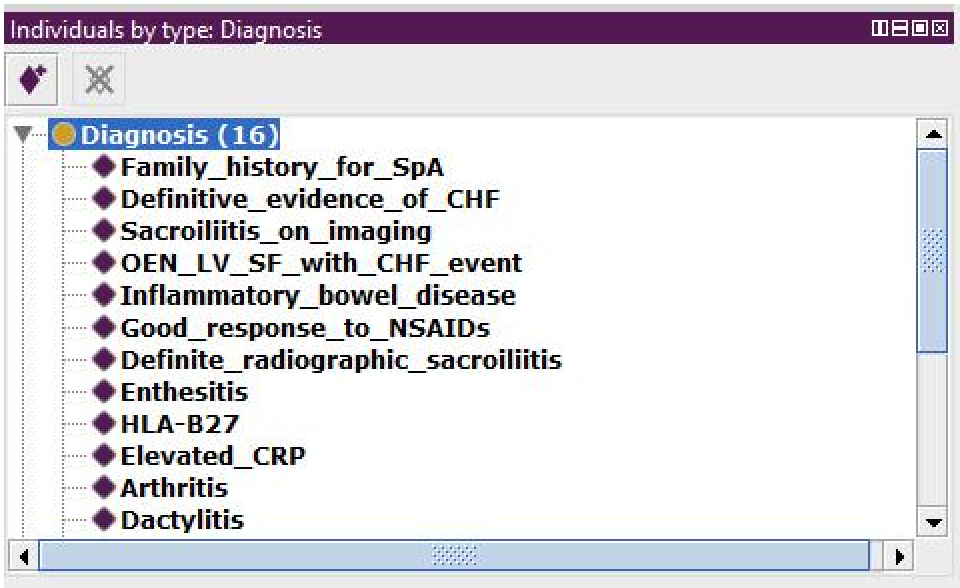

Figure 3.6: Some example individuals under class "Diagnosis".

There are altogether 62 individuals created. By generating classes and individuals, we could represent interested objects in the domain, thus implement classification methods on these objects according to their characteristics.

3.3.3 Object Properties of the Ontology

Object properties describe relationships between different classes, and each object property contains one domain and one range. Figure 3.7 shows all object properties we used in this system.

Figure 3.7: Object properties of the system.

has_diagnosis is the object property owned by class *Patient*. With domain Patient and range Diagnosis, this object property defines the relationship between patients and different diagnosis. The usage of this object property is introduced in Figure 3.8.

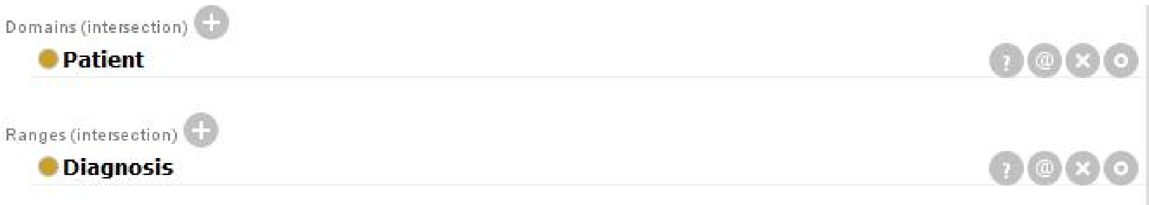

Figure 3.8: Domain and range of has_diagnosis.

has_disease is another object property owned by class Patient. It indicates the relationship between class *Patient* and class *Disease*. The domain and range of *has_disease* is shown in Figure 3.9.

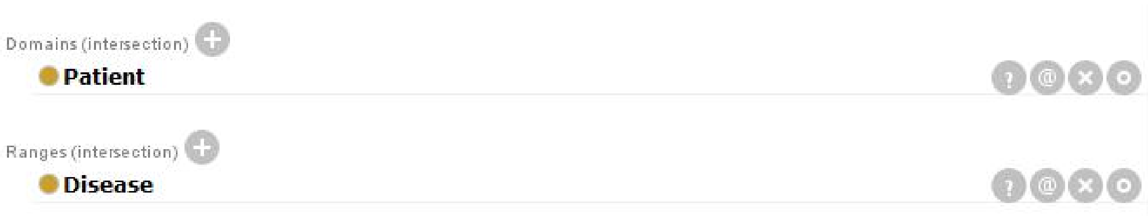

Figure 3.9: Domain and range of has_disease.

indicates is an object property owned by class *Diagnosis*. In clinical environments, medical experts like doctors may diagnose patients' characteristics according to domain knowledge and medical terms, and then provide patients with diagnosis such as which disease patients may have. Thus, a collection of diagnosis could indicate certain diseases for patients. The domain and range of object property *indicates* are shown in Figure 3.10.

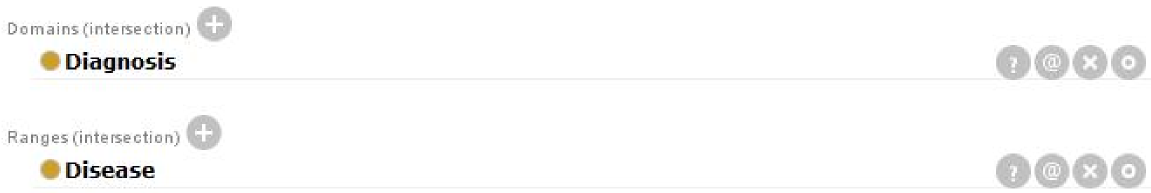

Figure 3.10: Domain and range of indicates.

presents is another object property owned by class *Patient*, and it provides the relationship between class *Patient* and *Signs_and_symptoms*. In real life, patients may present certain signs and symptoms as clinical manifestations. For example, in the disease SpA, patient may present "past or present spontaneous pain or tenderness at examination at the site of the insertion of the Achilles tendon", and doctors could then make diagnosis "enthesitis" for the patient, according to ASAS criteria. Figure 3.11 shows the domain and range of object property presents.

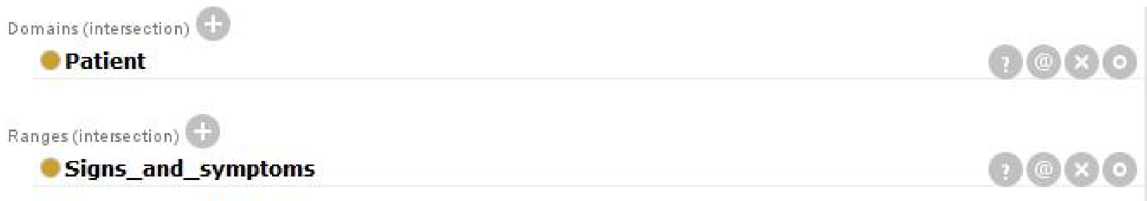

Figure 3.11: Domain and range of presents.

3.3.4 Data Properties of the Ontology

Data properties save data values for individuals in domain ontologies, and build up links between individuals. In this subsection, we present all data properties we used in this ontology.

Figure 3.12 shows all 6 data properties we used. Domain of all data properties was individual *Patient*, and range includes *string*, *integer* and *float*. In this subsection we will introduce data properties respectively.

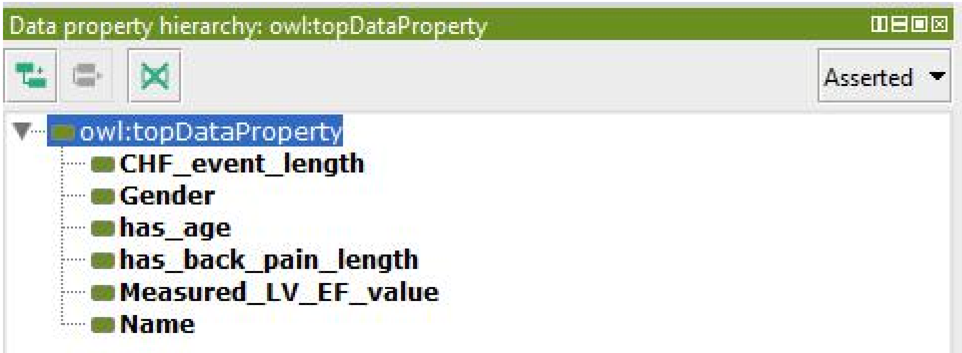

Figure 3.12: Data properties of the system.

CHF_event_length indicates the length of congestive heart failure event in DHF classification schema. This parameter was introduced in Table 3.2-Table 3.5, and its range is created as *integer*, with time unit “hour”.

Gender gives information about whether one patient is male or female. The range of data property *Gender* is *string.*

Has_age provides numerical information about patients’ age onset, and its range is *integer*, with unit “year”.

Has_back_pain_length is one data property concerns about observation information on patients about their length of having the symptom *back_pain.* In ASAS, back pain length is considered as one of important sufficient factor to evaluate if one person is potential SpA patient or not. The range of this data property is *integer*, and the unit is “month”

Measured_LV_EF_value is another numerical parameter in DHF classification schema, which stands for left ventricular ejection fraction (LVEF) value. Its range is *float.* Also defined from Table 3.2-Table 3.5, if one patient presents LVEF value no less than the threshold 0.50, he or she could be diagnosed as having *Objective_evidence_of_normal_LV_systolic_function.*

Name denotes patients' first name, with range *string*.

3.3.4 Patients Data Input

After creating all the classes and individuals, we manually imported patients' data as input to the system. These input data included information about patients' clinical characteristics like signs/symptoms, age, gender, LVEF value and back pain length .etc. Figure 3.13 shows the input data of Patient 1.

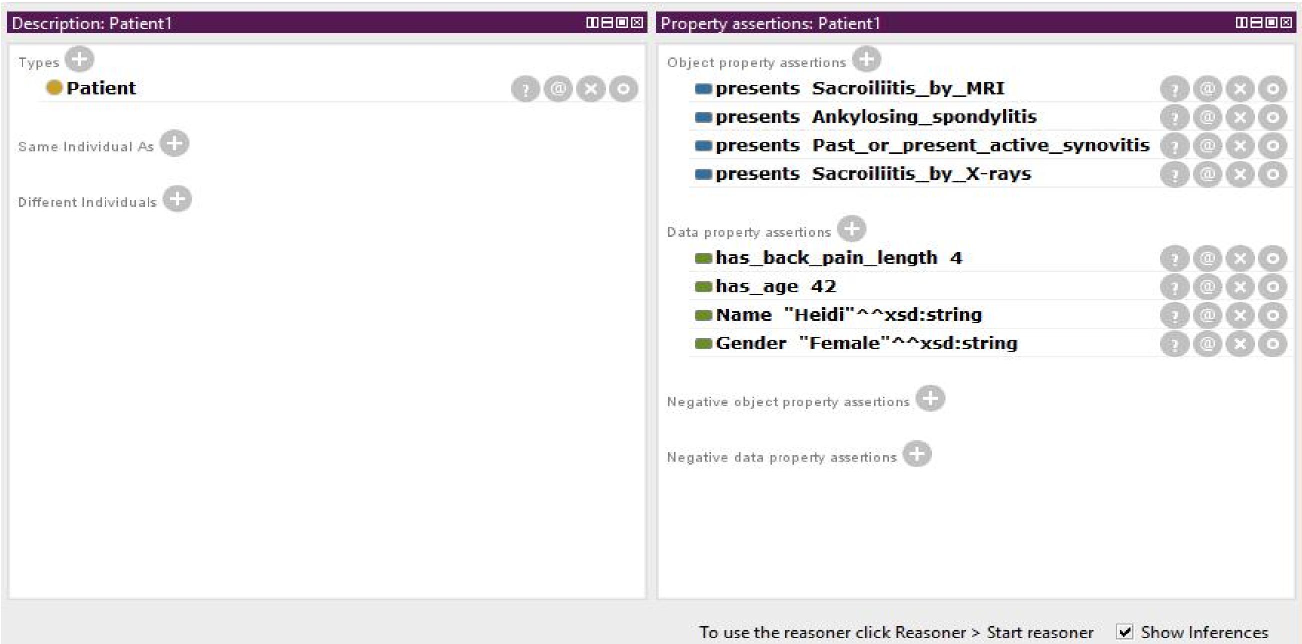

Figure 3.13: Input data of Patient 1.

Patient 1 has back pain length for 4 months and with age at onset 42. Her name is Heidi and gender is female. These data were imported into the system as data properties. While for clinical signs and symptoms, Heidi presents Scroiliitis by MRI, Ankylosing spondylitis, Past or present active synovitis and Scroiliitis by X-rays. This part of data were inserted through object properties.

Figure 3.14 shows another example of patient. In this set of imported data, patient 8 presents more clinical signs and symptoms than patient 1.

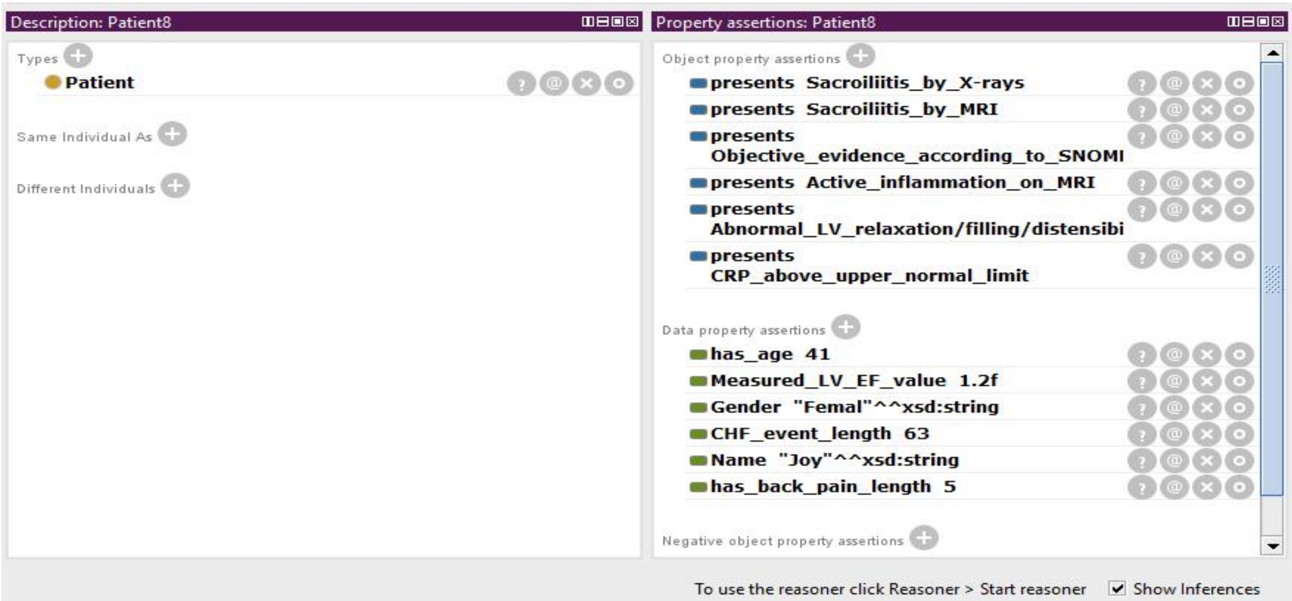

Figure 3.14: Input data of Patient 8.

In this way, 8 patients' data is imported into the system as input. We use these input data to make diagnosis and disease identification for patients, through different ontology reasoning approaches.

3.4 Summary of This Chapter

In this chapter, we illustrated the framework architecture of our ontology and CDSS. Related efforts on how to develop medical ontology and CDSS were presented. In section 3.1, we showed an abstract overview of our ontology model architecture, and OWL was selected as the basic tool in this development work, as it could provide a rich set of knowledge representation functions, such as classes, individuals and properties. We also discussed relevant terminological sources in this section. These sources include ASAS criteria and classification schema for DHF. In section 3.2, ontology-associated generic tools were briefly introduced, and we decided to choose Protégé as the modeling tool, while OWL-2 language as the modeling language. Our efforts in developing the ontology and CDSS were shown thereafter. As a result, in total 23 classes, 62 individuals, 4 object properties and 6 data properties were created to structure the ontology. This information was given in section 3.3. Through this ontology development process, we could map data from hospital information system and biomedical terms/terminologies to the model. Since reasoners in Protégé could check consistency of the ontology, it is legitimate to use them to guarantee the rationality and validity of the ontology.

In next chapter, we will demonstrate how we implemented ontology reasoning through SWRL, OWL DL and SPARQL. Since the objective of reasoning on this ontology is to output diagnosis for patients, as well as identify diseases according to patients' characteristics, we will investigate whether these three semantic web-based techniques are appropriate for supporting clinical decisions.

Chapter 4: Ontology Reasoning

Ontology reasoning is a suitable approach to provide essential services and decision support in many applications. For applications in which ontologies perform key role, ontology reasoning could help with deriving hidden patterns and facts that are not expressed explicitly in ontologies and knowledge base. Ontology reasoning techniques are widely used in various aspects of real world, ranging from business applications to diagnostic facts finding.

In this chapter, we demonstrate three ontology reasoning approaches that were implemented in this CDSS: SWRL, OWL DL and SPARQL. We aim to investigate whether these reasoning techniques are feasible and legitimate to assist with disease identification and patient classification. We use the ontology introduced in Chapter 3, and apply three reasoning methods on classes and individuals.

4.1 Reasoning via SWRL

SWRL is widely used as a main reasoning technique in semantic web. As the main goal of semantic web is to provide interoperability for the CDSS, SWRL is one key approach which could achieve this goal. The following list describes the different steps in the reasoning approach via SWRL, from rule construction to rule implementation. The results of SWRL reasoning will be stored in the ontology and CDSS.

1. Rule construction. We construct production rules in Protégé, which is the most widely used ontology development platform. In this software, we could use the highly-interactive and full-featured SWRLAPI to construct rules with IF-THEN form. After writing rules, we could take advantage of plugin mechanism in Protégé-OWL, to integrate our production rules into third party rule engines such as Drools [44].

2. Rule implementation. After the production rules are created, we use them to perform reasoning. SWRLAPI supports an OWL profile called OWL 2 RL and uses an OWL 2 RL-based reasoners to perform reasoning. The reasoners include Fact++, Hermit and Pellet, etc. The rule engines in

SWRLAPI could help to transfer the inferred rule engine knowledge to OWL knowledge, in order to execute reasoning to the ontology which is created in OWL files.

3. Results presentation and storage. As reasoners assisted to provide diagnostic classification services, the results which contain information about automatic disease diagnosis and disease identification could be presented as output of the system. These results were presented as subclasses or property assertions in Protégé-OWL, and stored in patients' profiles in OWL format.

4. Results evaluation. Medical terminologies and terms were engaged to help with checking the correctness of the ontology, in terms of domain concepts representation and symptoms-to-diagnosis/diagnosis-to-disease relationships. We used these reasonable references to confirm the validity of the output results.

Since the outcome of reasoning process generally deals with matters of computer-aided medical and clinical heath care, it is vital important to ensure that the outcome is identical to diagnostic and biomedical terms and criteria. In other words, the results of the CDSS should coincide with medical knowledge which is widely agreed and recognized in society.

4.1.1 Rule Construction

SWRLAPI is used to write Horn-like rules. These rules are constructed to express OWL concepts in the ontology, and reason about OWL individuals. Hidden patterns and new knowledge are inferred from existing knowledge base, and in this case, diagnosis and disease information is deduced.

Figure 4.1 displays the API of SWRL Tab in Protégé. In this API, we could use the SWRL editor to write rules with IF-THEN logic. This open-source rule editor enables us to seamlessly integrate SWRL rule knowledge with OWL knowledge, and it is quite flexible to switch between rule editing and OWL editing. Supported by a subsystem called Protégé SWRL Factory, SWRL editor permits high-level interoperability between SWRL and rule engines. In this editor, we could write rules with use of domain knowledge, and these rule information will be automatically and tightly integrated with Protégé OWL.

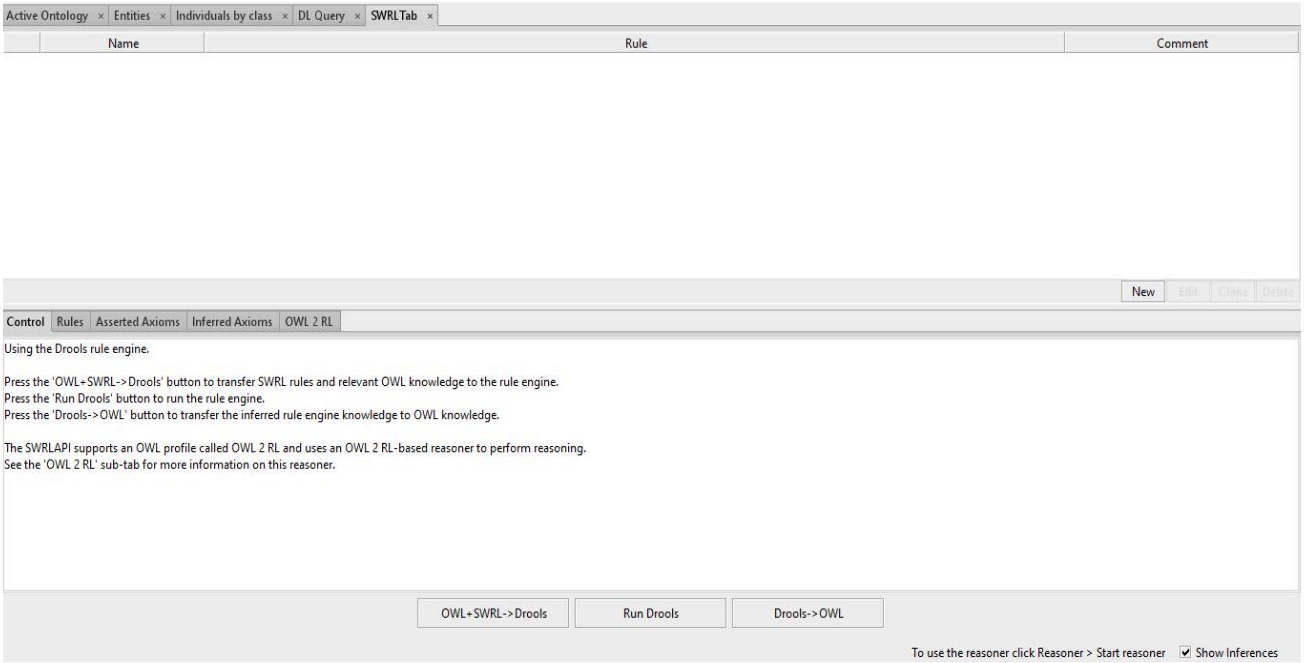

Figure 4.1: Screenshot of SWRL Tab in Protégé.

With the assist of SWRLAPI, we create 84 production rules, and these rules are denoted with names and comments. Figure 4.2 shows the collection of SWRL rules. It should be noted that we create these rules according to SNOMED CT, ASAS criteria and DHF classification schema.

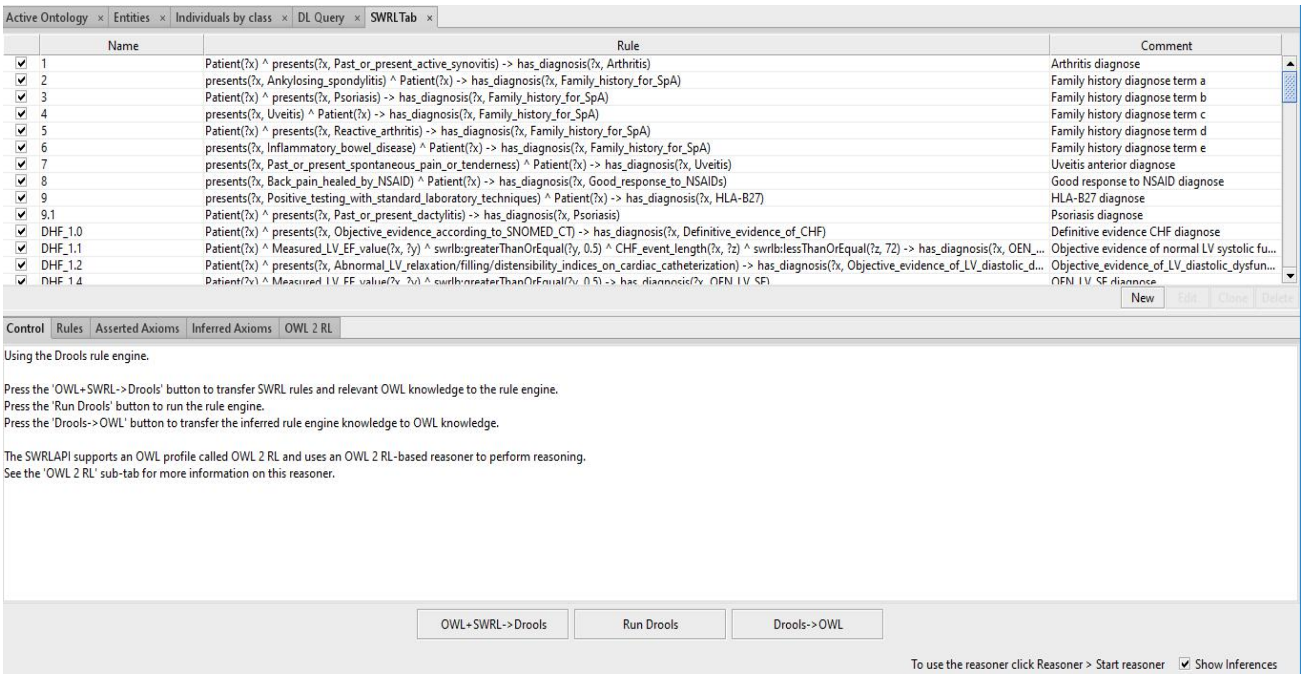

Figure 4.2: Screenshot of created production rules in SWRL Tab.

4.1.2 Rule Implementation

The next step is to implement these created production rules. As a plug-in to the SWRLAPI, SWRLAPI Drools engine supports the execution of rules, and it also enables OWL 2 RL-based reasoners to implement classification tasks in Drools. In SWRL Tab, there is a bridge mechanism to provide interaction and connection between OWL knowledge base and SWRL rules, and we could evoke Drools engine as an intermediate to achieve this transform. Figure 4.3 demonstrates the results we got after running Drools rule engine in SWRL Tab.

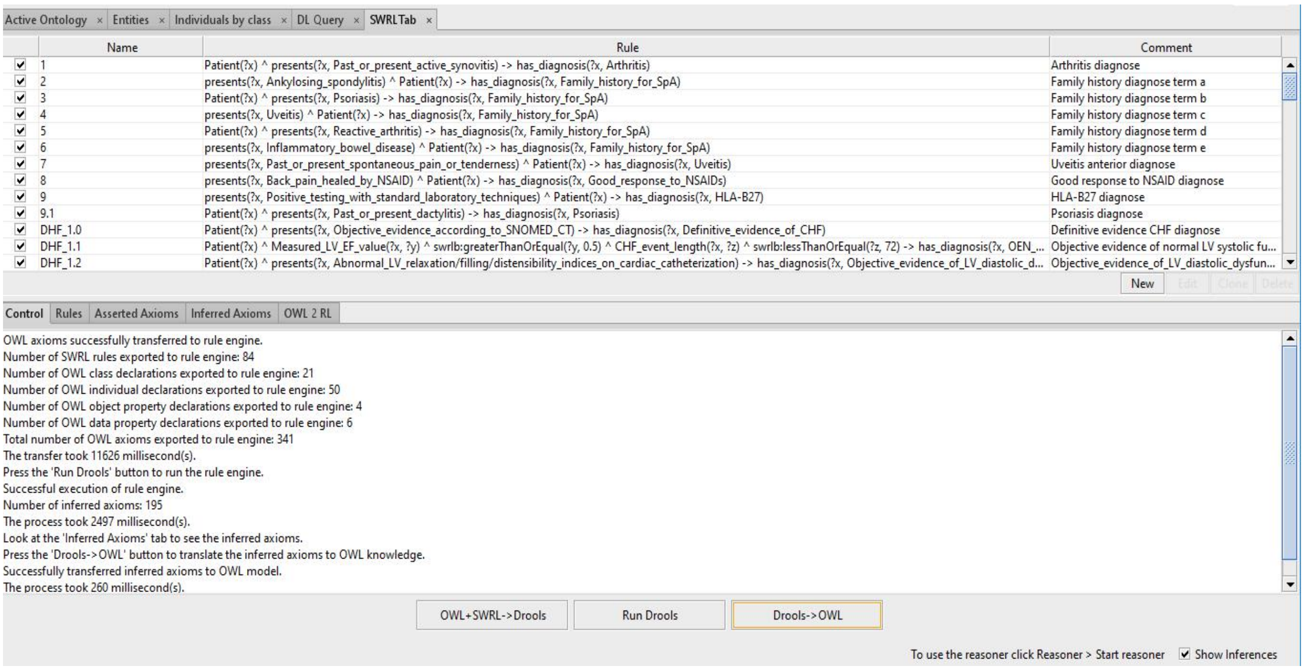

Figure 4.3: Transforming process of inferred axioms through Drools engine.

After clicking the button "OWL+SWRL->Drools", SWRL rules and relevant OWL knowledge are transferred to the Drools engine. Afterwards, Drools engine is executed to transfer the inferred rule engine knowledge to OWL knowledge. By this process, we achieve seamless switch between SWRL rule editing and OWL editing of classes, individuals, object properties and data properties in our ontology. From Figure 4.3 we can observe that OWL axioms are successfully transferred to rule engine, and all 84 SWRL rules are correctly exported to Drools.

4.1.3 Results Representation and Storage

Since the rules are successfully implemented, we could collect results as output of the system, and represent them automatically by running reasoners. The role of reasoners is to check the consistency of ontologies, thus provide classification services. In our system, we have loaded the ontology, and determined all axioms in it. When SWRL rules are constructed, it means that we have resolved all imports of the system, and also parsed the contents of the OWL file. Thereafter, reasoners could help to check the consistence of the ontology in the following aspects [50]:

- Reasoners check whether there exists a relational structure that matches all axioms in the ontology.
- Reasoners test whether there exists a relational structure in which all the instances of axioms matches with the individuals in the created ontology.
- Reasoners test whether two arbitrary classes have relationship “subclass” with each other.

The results of these detection and tests will show in the ontology OWL file, and they are easy to read by users.

Figure 4.4 shows results of diagnosis and disease identification of Patient 1. Note that the texts in bold types are the input which were manually inserted to the system, and texts with normal font, as well as colored background, are output of the system. We could observe that with presenting *Sacroiliitis_by_MRI, Ankylosing_spondylitis, Past_or_present_active_synovitis* and *Sacroiliitis_by_X-rays* as symptoms, Patient 1who has back pain length for 4 months, and with age at onset 42, has been classified under the class *Possible_SpA_patient*, and he has been diagnosed as *has_disease SpA*.

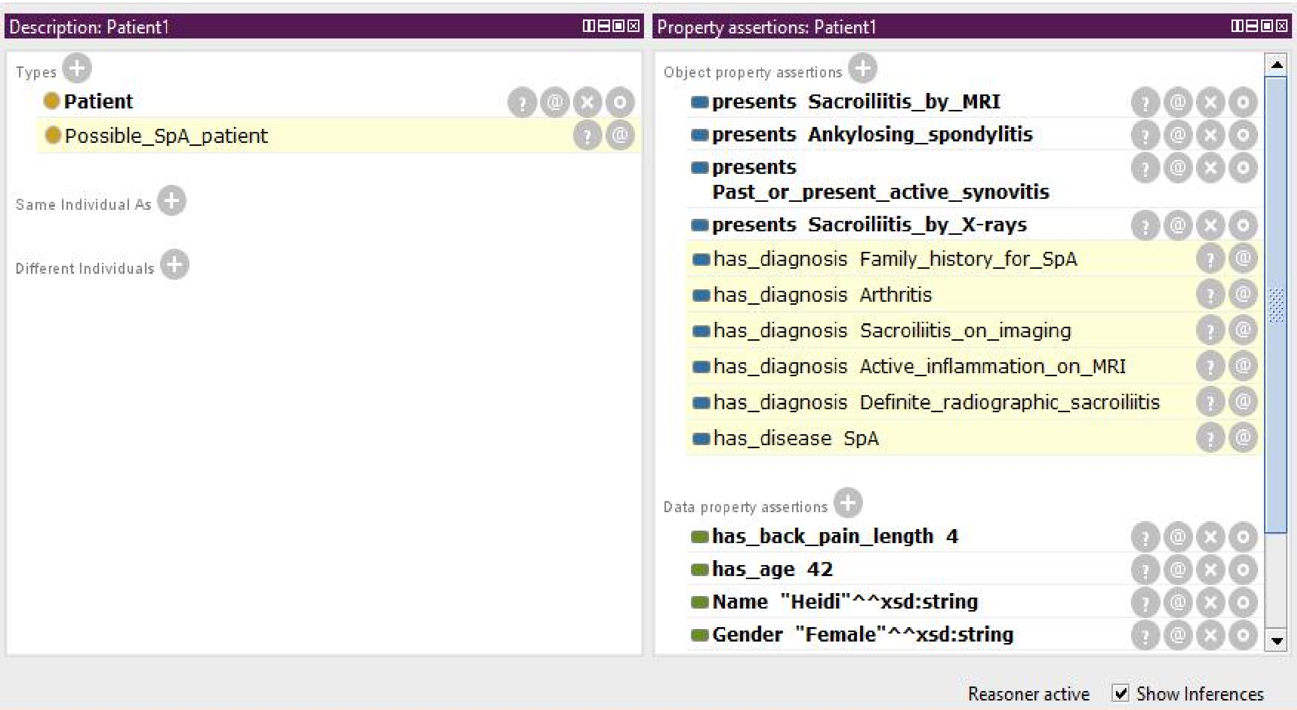

Figure 4.4: Diagnosis and disease identification for Patient 1.

We give another patient example. Figure 4.5 shows another patient classification results. Patient 5 is a female who was measured LVEF value with 0.6 and CHF event length for 70 months. She also has had back pain for 7 months, and she is at onset 36 years old. With presenting *Objective_evidence_according_to_SNOMED_CT* and *No_conclusive_information_on_LV_diastolic_fu-nction* as clinical symptoms, she has been classified under classes *Possible_DHF_patient*, *Possible_SpA_patient* and *Probable_DHF_patient*. In the right side, the system made diagnosis *OEN_LV_SF*, *Definitive_evidence_of_CHF*, *OEN_LV_SF_with_CHF_event* and *Objective_evidence_of_LV_diastolic_dysfunction_is_lacking*.

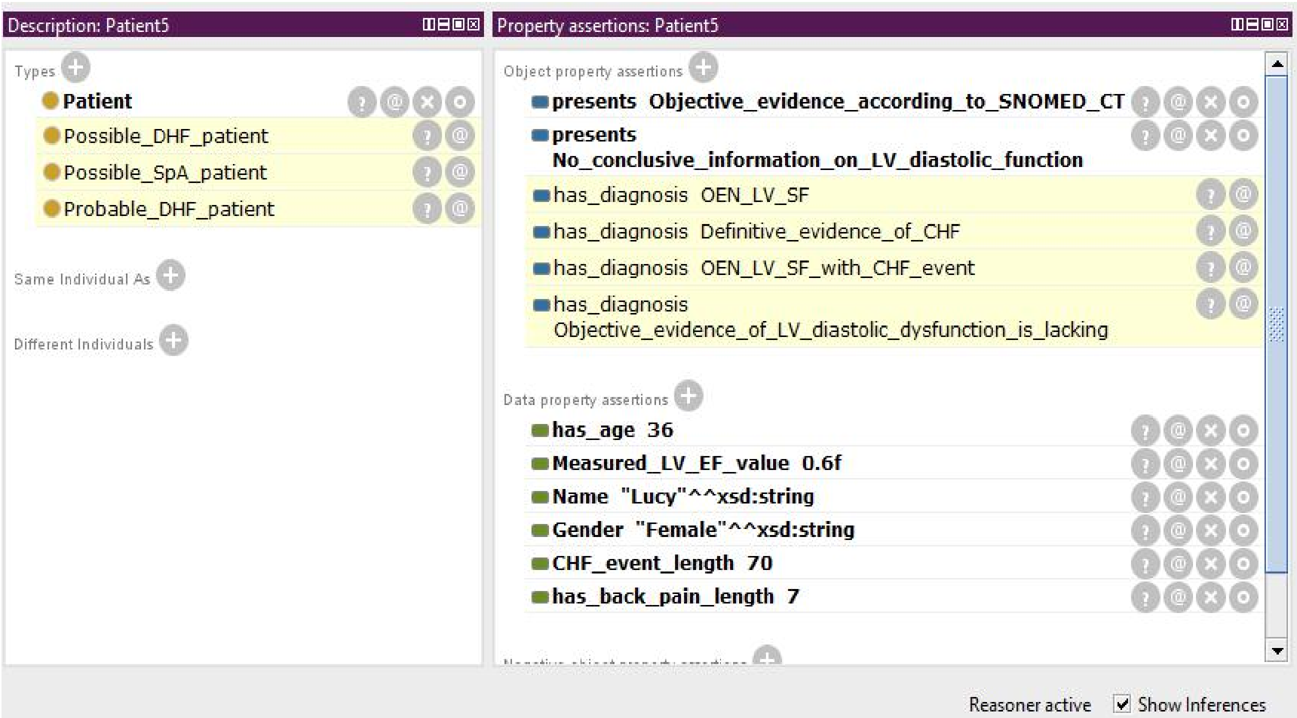

Figure 4.5: Diagnosis and disease identification for Patient 5.

For Patient 2 who is 39 years old, clinical manifestations *Insidious_onset*, *Improvement_with_exercise* and *No_improvement_with_rest* are presented, and he has back pain length for 2 months. After running the reasoner, he has been diagnosed as having the disease IBP. However, since patients should meet the sufficient condition of having back pain for more than 3 months, Patient 2 is excluded from the subset *Possible SpA patient*. The diagnosis and disease identification results of Patient 2 are explicated in Figure 4.6.

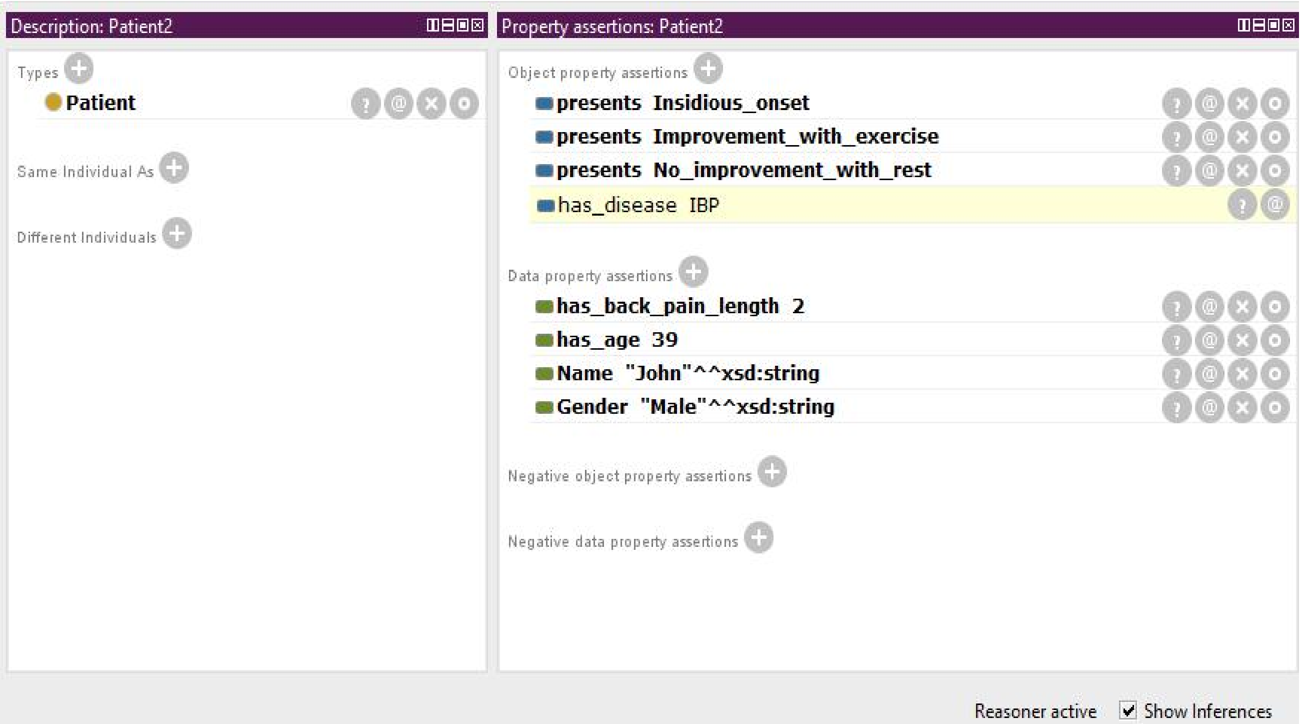

Figure 4.6: Diagnosis and disease identification for Patient 2.

With this methodology, we collect results for all 8 patients. Reasoners showed that our ontology reasoning process was successful to generate output data.

4.1.4 Results Evaluation

It is vital important to assess and evaluate the correctness and degree of accuracy of results. In this subsection we examine how these rules aligned to produce output as diagnosis and disease identification. Due to the large quantity of rules, we select 5 representative pieces out of 84 rules, and investigate how they work in inference engine.

Rule 1: Patient(?x) ^ presents(?x, Past_or_present_active_synovitis) -> has_diagnosis(?x, Arthritis)

This simple rule builds the relationship between sign/symptom *Past_or_present_active_synovitis* with diagnosis *Arthritis.* This rule is translated from one ASAS term, and we use this axiom to test if any patient has the clinical symptom *Past_or_present_active_synovitis.* Figure 4.7 shows the user interface of SWRL editor where we could construct rules, and Figure 4.8 shows how this rule helps to produce output.

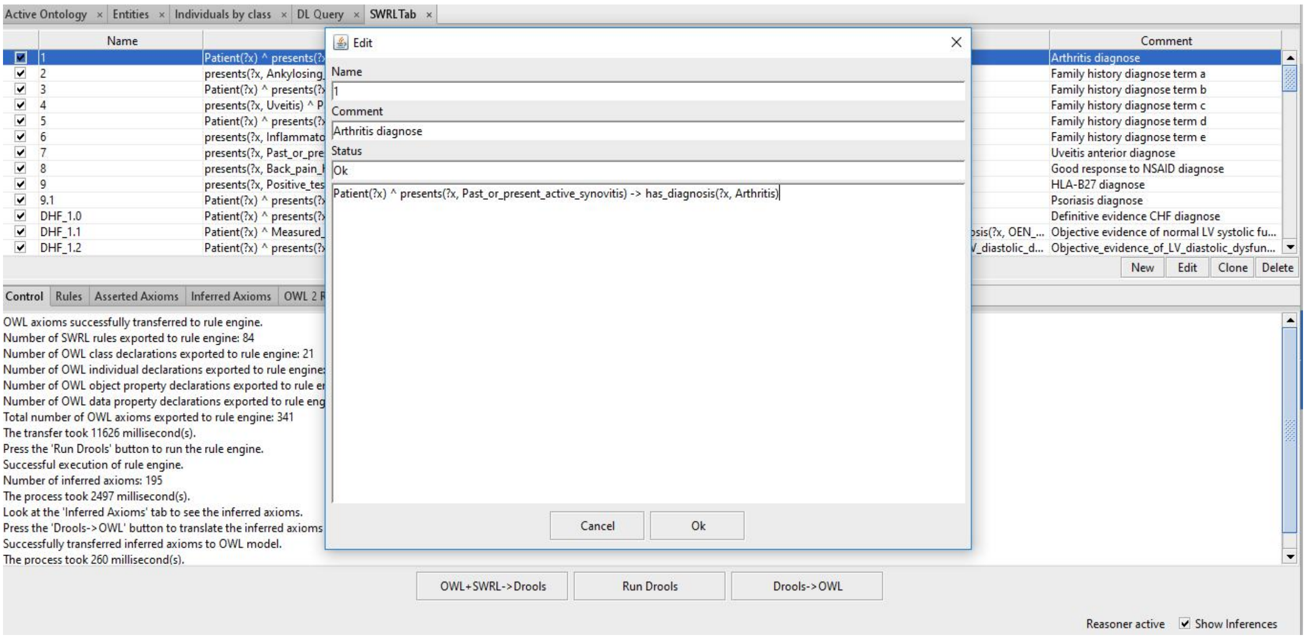

Figure 4.7: Rule 1 construction user interface in SWRL editor.

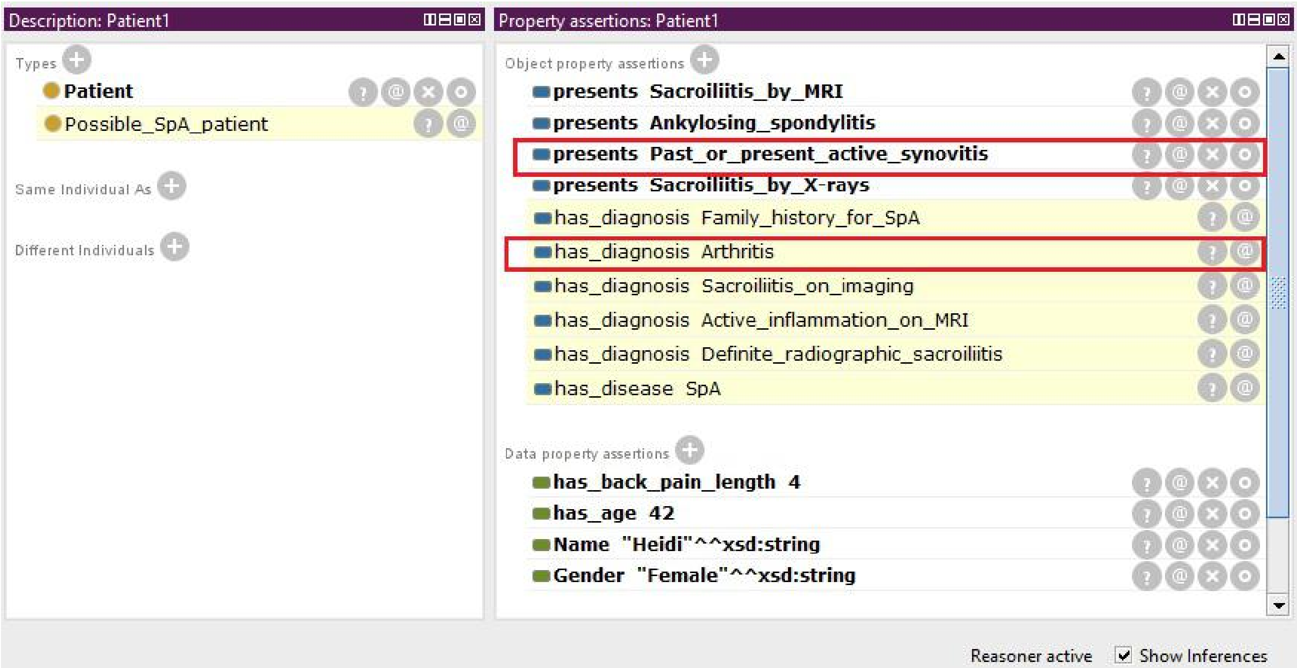

Figure 4.8: Result of Rule 1.

This rule uses IF-THEN logic to determine whether the system should has diagnosis Arthritis. Since Patient 1 presents the clinical sign/symptom *Past_or_present_active_synovitis*, the decision is made correspondingly.

Rule 2: presents(?x, Sacroiliitis_by_MRI) ^ presents(?x, Sacroiliitis_by_X-rays) ^ Patient(?x) -> has_diagnosis(?x, Sacroiliitis_on_imaging)

This rule demonstrates the diagnosis *Sacroiliitis_on_imaging*. Logical symbol "^" denotes the logic "and". Thus, this rule means: if one patient x presents clinical signs/symptoms *Sacroiliitis_by_MRI* and *Sacroiliitis by_X-rays*, then the system should give the diagnosis *Sacroiliitis_on_imaging* to the patient. Figure 4.9 and Figure 4.10 demonstrate Rule 2 construction user interface in SWRL editor and the corresponding results.

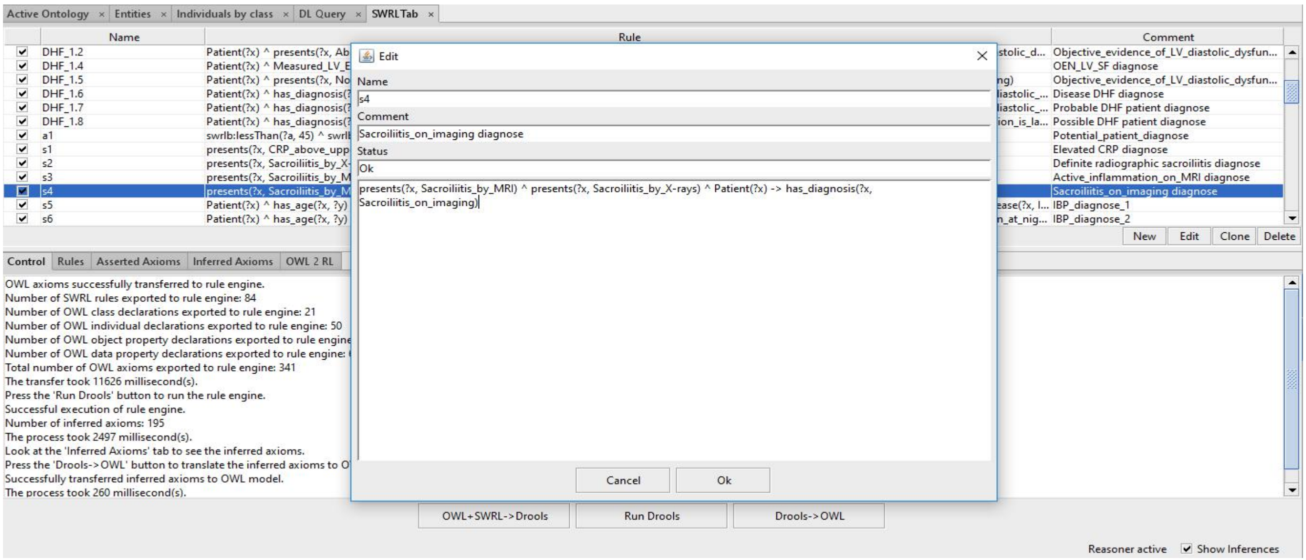

Figure 4.9: Rule 2 construction user interface in SWRL editor.

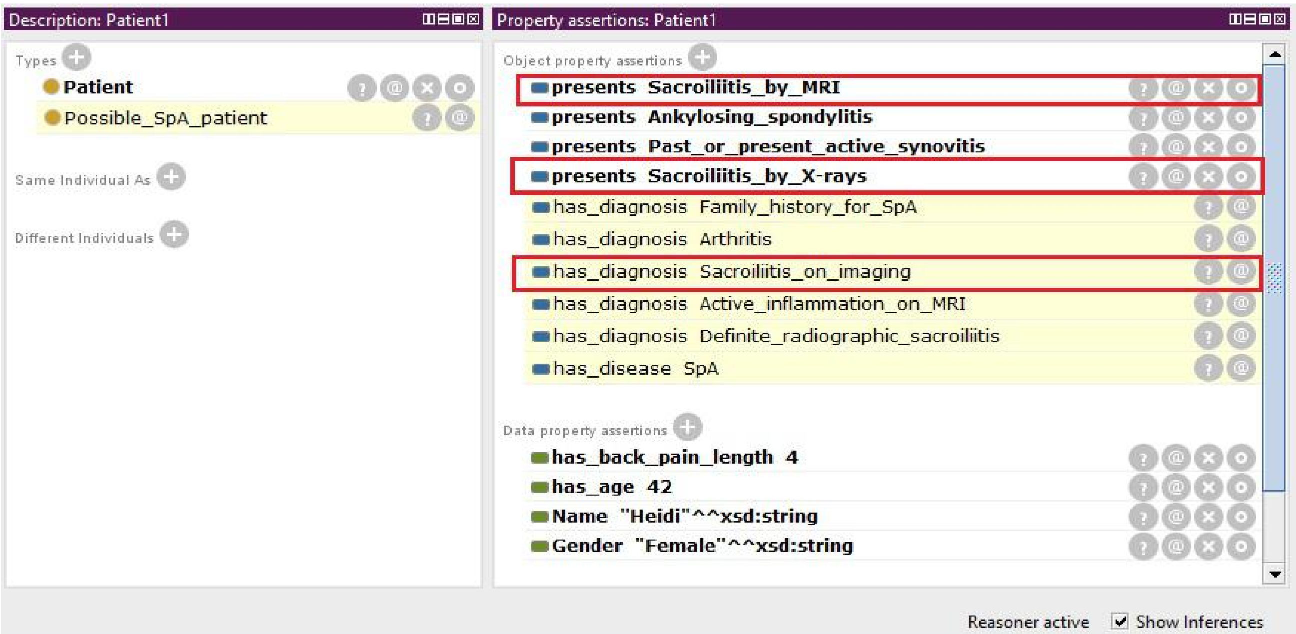

Figure 4.10: Results of Rule 2.

Rule 3: Patient(?x) ^ has_diagnosis(?x, Sacroiliitis_on_imaging) ^ has_diagnosis(?x, Arthritis) -> has_disease(?x, SpA).

As seen in chapter 3.1, disease SpA could be diagnosed via either Sacroillitis on imaging approach, or HLA-B27 approach. In consideration of this medical term in ASAS, this rule is constructed to make disease identification for a patient who has already been diagnosed with Sacroiliitis on imaging. It should be noted that this rule is one of 57 rules that are intended to handle diagnosis concerning of disease SpA. Figure 4.11 and 4.12 show the construction and results of this rule.

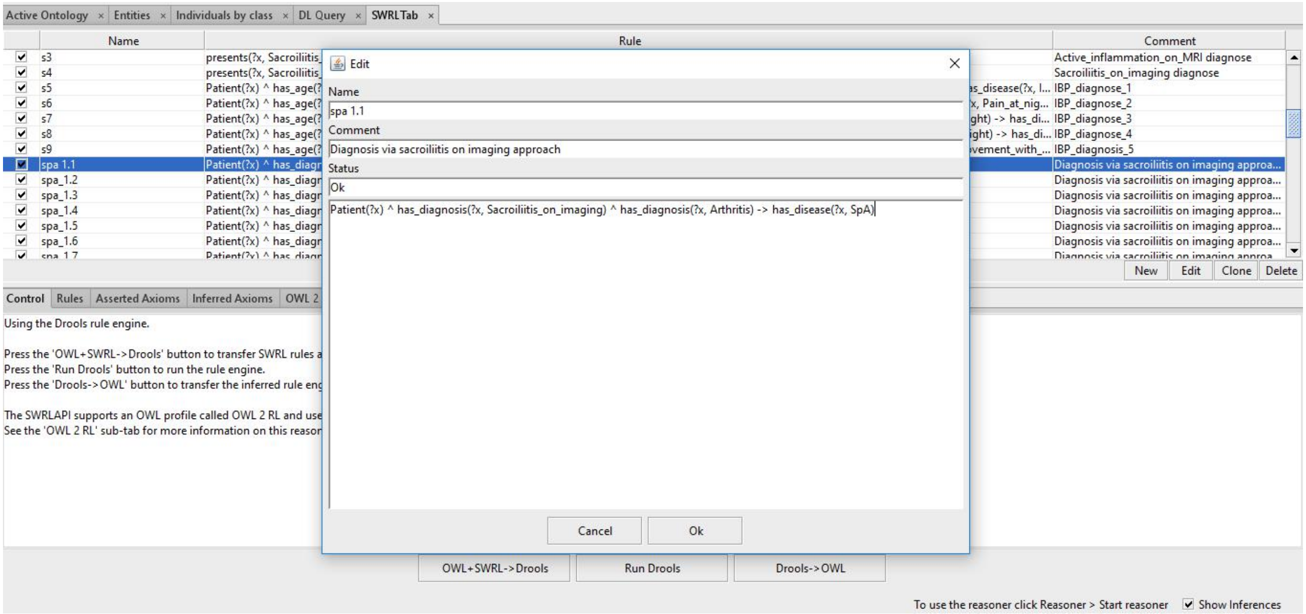

Figure 4.11: Rule 3 construction user interface in SWRL editor.

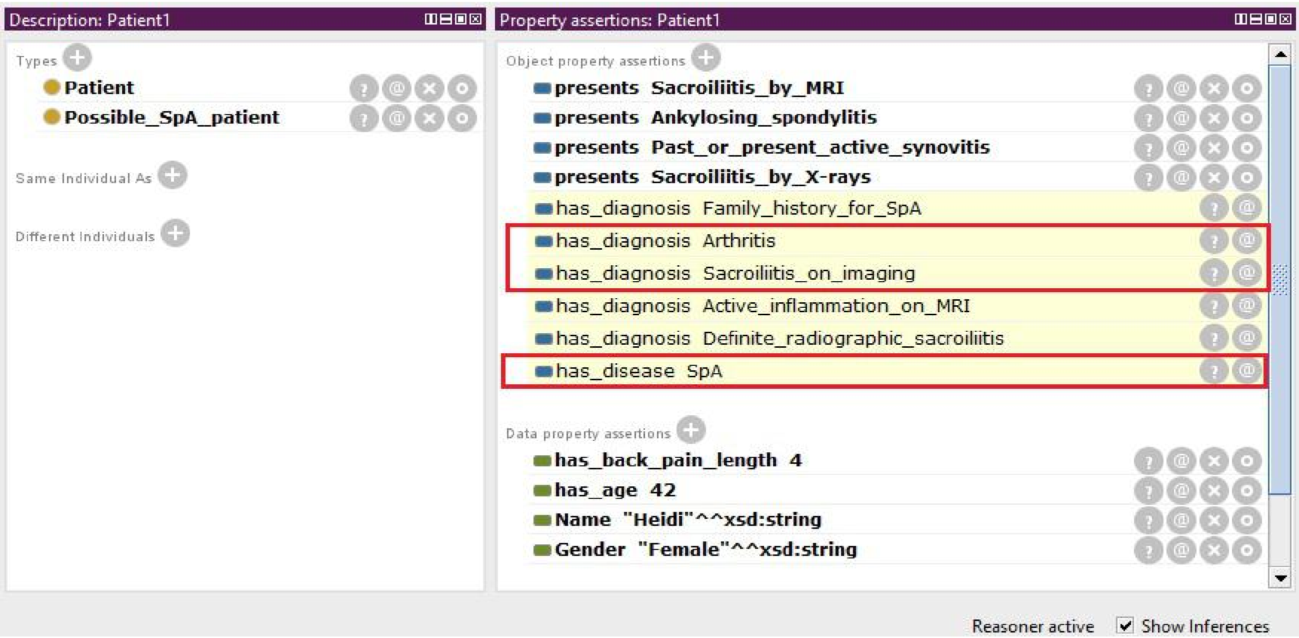

Figure 4.12: Results of Rule 3.

Rule 4: Patient(?x) ^ Measured _LV_EF_value (?x,?y) ^ swrlb:greaterThanOrEqual(?y,0.5) ^ CHF_event_length (?x,?z) ^ swrlb:lessThanOrEqual (?z,72) -> has_diagnosis (?x,OEN_LV_SF_with_CHF_event).

This rule takes use of SWRL build-ins. Build-ins in SWRL are used to further extend the functions of rule language to support advanced taxonomy. In this case, swrlb:greaterThanOrEqual and swrlb:lessThanOrEqual are two build-in atoms to support numerical comparisons. To accomplish this goal, numerical values are imported to the system as data properties of individuals. Figure 4.13 and 4.14 show how this rule works on Patient 5.

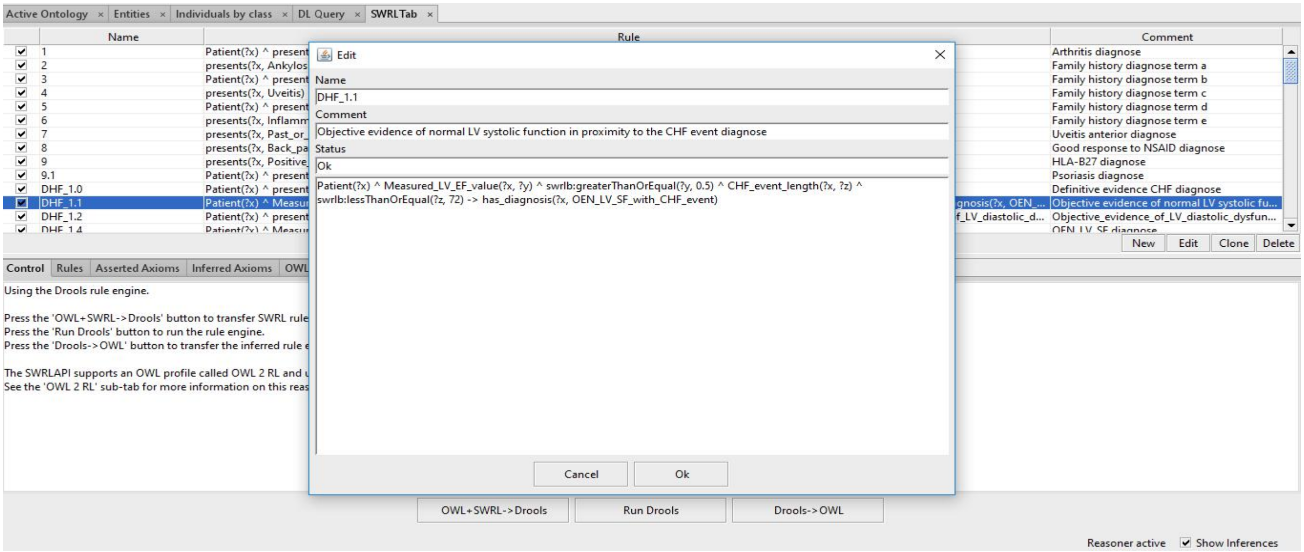

Figure 4.13: Rule 4 construction user interface in SWRL editor.

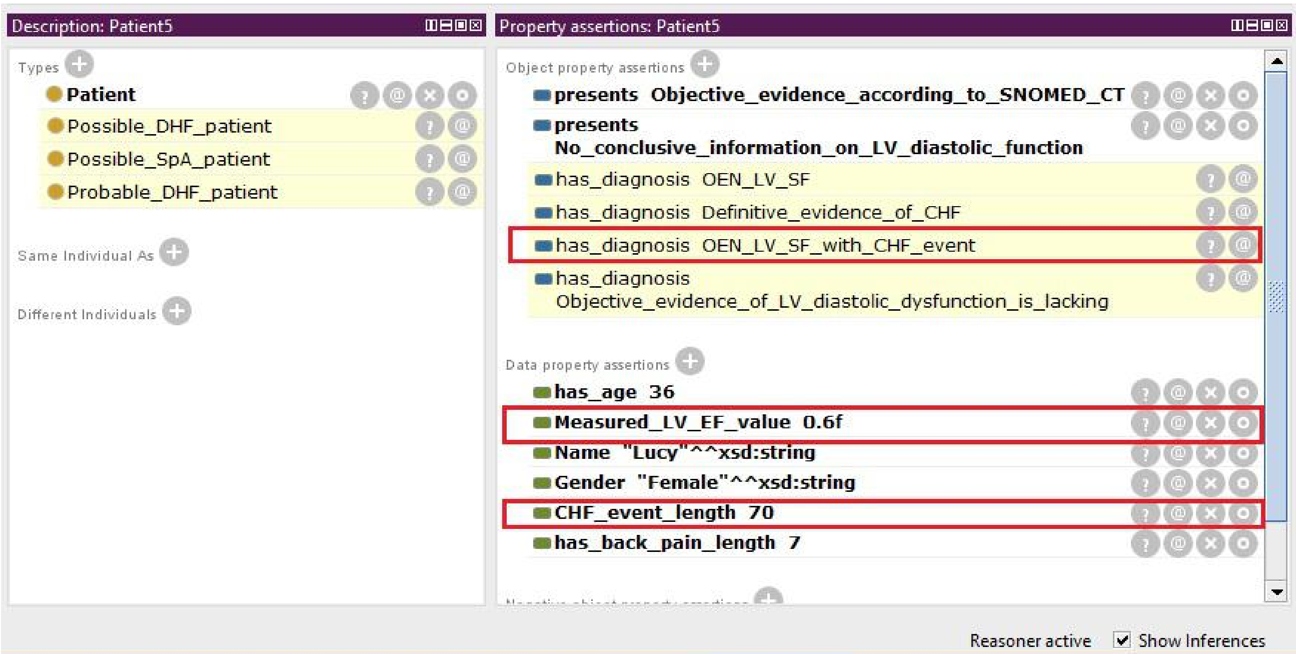

Figure 4.14: Results of Rule 4.

Rule 5: swrlb:lessThan(?a, 45) ^ swrlb:greaterThan(?y, 3) ^ Patient(?x) ^ has_back_pain_length(?x, ?y) ^ has_age(?x, ?a) -> Possible_SpA_patient(?x).

Rule 5 settles the issue of identifying whether one patient has the potential to be sickened by SpA. In ASAS diagnostic criteria, one person could be diagnosed to be potential SpA patient if he or she is under 45 years old, and has had pain back for more that 3 months. Therefore, we use data properties to classify patients. Notice the patient is classified under the subclass *Possible_SpA_Patient*.

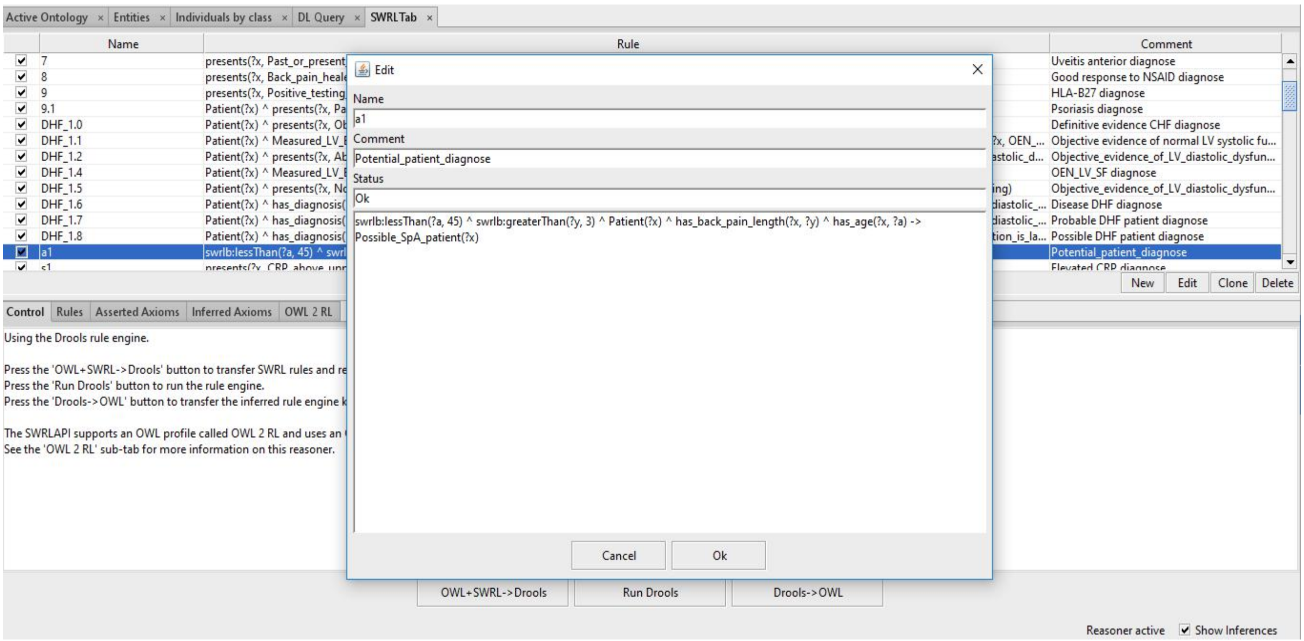

Figure 4.15: Rule 5 construction user interface in SWRL editor.

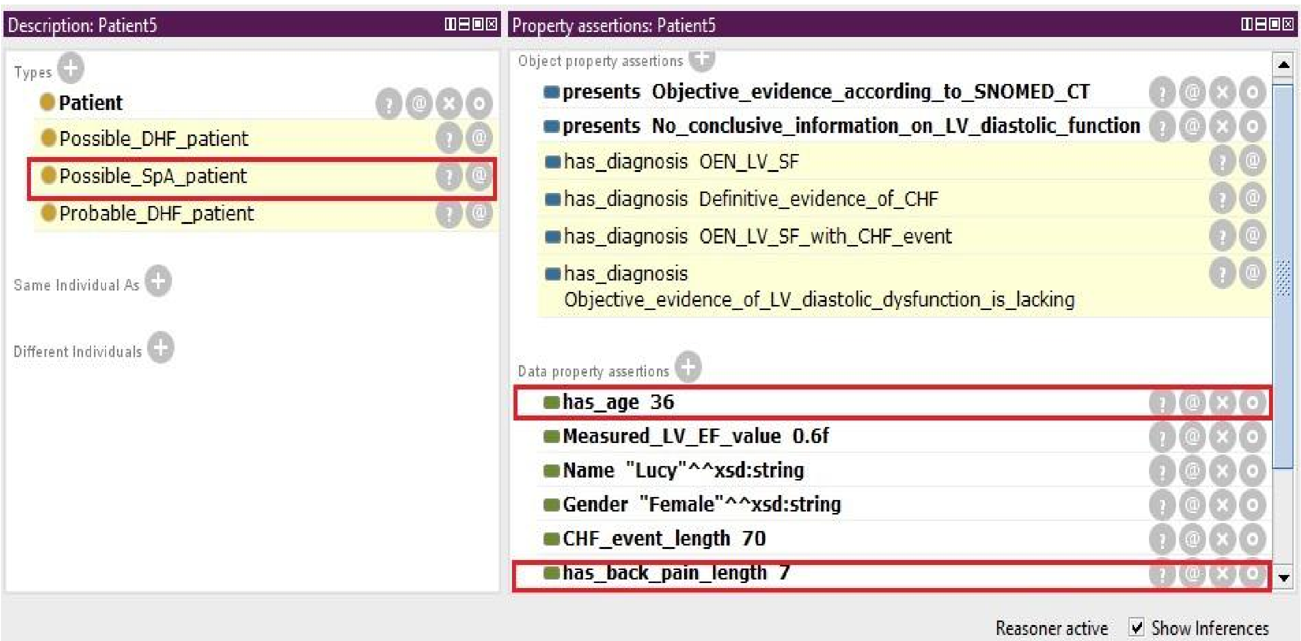

Figure 4.16: Results of Rule 5.

The ontology reasoning via SWRL enabled data to be mapped from biomedical terminologies and hospital information system to the ontology and CDSS model. In this model, reasoners run to check the consistency of the ontology, and patients are classified based on several clinical factors, such as back pain length, age, LVEF value, CHF event length, etc. In this approach, the ontology reasoning is implemented on individuals in the CDSS.

4.2 Reasoning via OWL DL

From [37], we learned that OWL DL is also a powerful technique for ontology reasoning. The authors represented diagnostic criteria of SpA via OWL DL, and successfully proved reasoners have capabilities to enable automated classification. In this section, we extend their work. In our CDSS model, we create extra classes for representing signs/symptoms not only in SpA, but also in disease IBP and DHF. We follow a similar methodology with previous section:

1. Ontology modeling. Instead of constructing rules in SWRL approach, we model all signs/symptoms, diagnosis and diseases in classes. OWL DL implements reasoning through class expression editor, which is primarily used to create a full range of class expressions in Manchester OWL Syntax. Therefore, we model all classes with class expressions including subclasses, disjoint classes, properties and class assertions.

2. OWL DL implementation. Since class expression editor in Protégé could automatically check the correctness of syntax, it ensures that our ontology model could fulfill the necessary conditions for ontology reasoning. Reasoner Hermit is selected to implement reasoning, and results will thereafter be collected.

3. Results presentation and storage. As the results are presented in class level, patients are classified under certain disease and diagnosis classes. For this reason, we do not have property assertions for patients in this approach. Instead, patients' information is presented via descriptive knowledge.

4. Results evaluation. We also used popular-used medical terminologies like SNOMED CT and ASAS Criteria to check the correctness of results. We will describe how we evaluate the results in next subsection.

4.2.1 Ontology Modeling

Based on the ontology created in chapter 3, we modify the structure according to properties of OWL DL. As mentioned in previous subsection, individuals were all transferred to classes in this approach. Figure 4.17 shows part of the class hierarchy in the new ontology. Note that all signs and symptoms are represented through classes, and this is the vital difference compared to the ontology in Chapter 3.

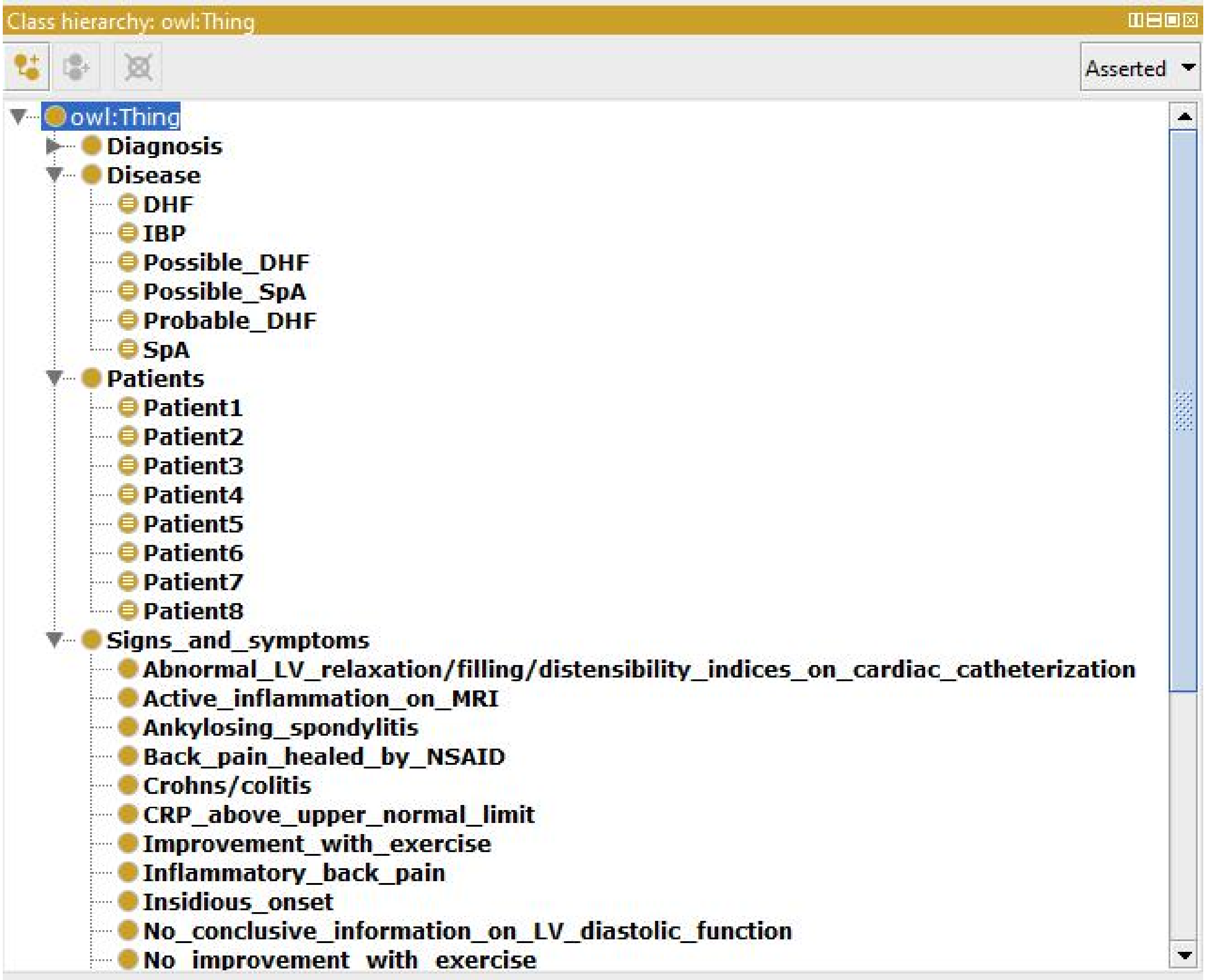

Figure 4.17: Part of the class hierarchy in OWL DL-based ontology.

Figure 4.18 and 4.19 show the object and data properties in the ontology. Considering that the diagnosis and disease classes perform the role as automatic facilities to find if there exists a corresponding sign/symptom, we use equivalent object property: *has_finding* = *presents*.

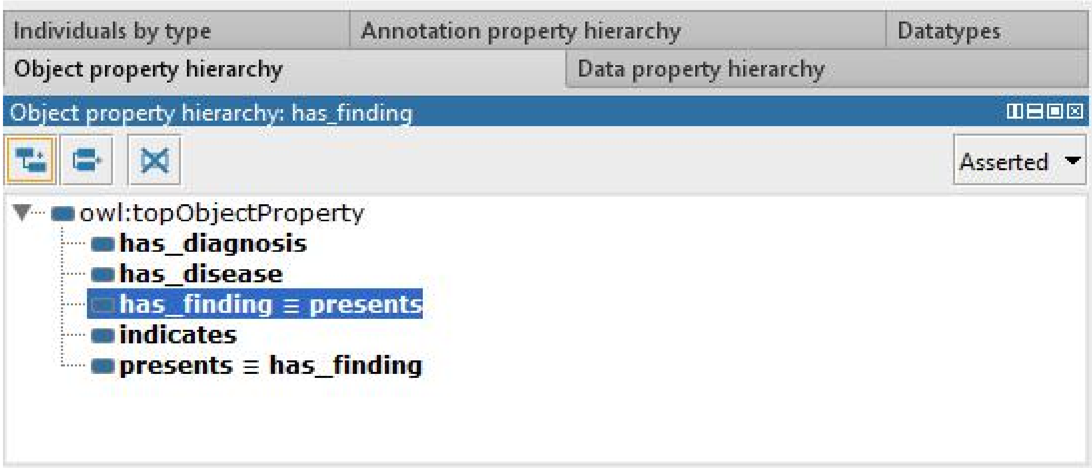

Figure 4.18: Object properties in OWL DL-based ontology. *Has_finding* and *presents* were created as equivalent properties.

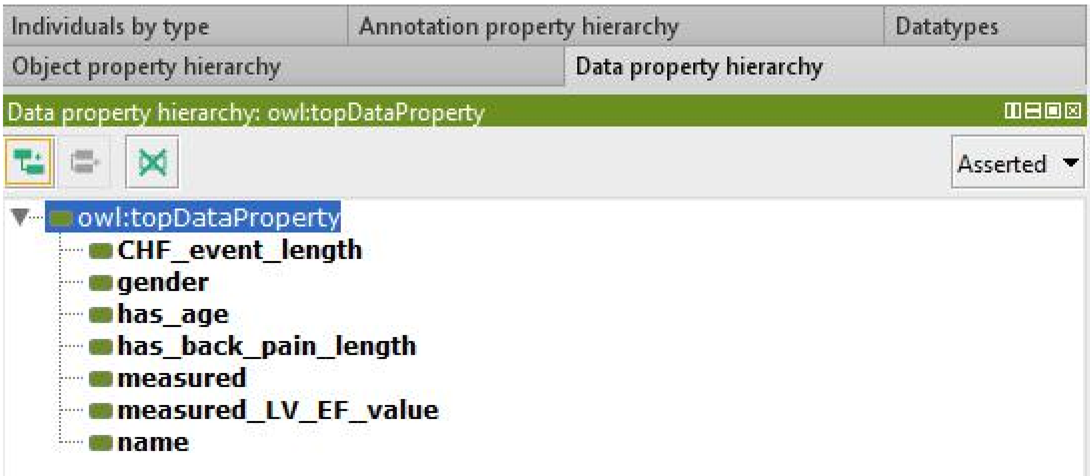

Figure 4.19: Data properties in OWL DL-based ontology.

4.2.2 OWL DL Implementation

Since Protégé uses the Manchester OWL Syntax in all expressions, we represent all diagnosis and disease classes in class expression editor. Object property *has_finding* indicates the operation of all classes to find if there exists any potential subclass. These potential subclasses could be deemed as hidden patterns in the ontology, before the reasoner is executed to implement ontology reasoning.

Figure 4.20 shows an example in which the diagnosis class *Arthritis* is issued. On the right side, Arthritis is described with descriptive knowledge in the class level. This syntax is written to find if there is any other class obtaining the same description *Past_or_present_active_synovitis*. If the ontology could find one, the derived class will be asserted under class *Patient_with_Arthritis*. Beside this, all other diagnosis classes are represented in a similar way, in order to implement automatic diagnose for all patients.

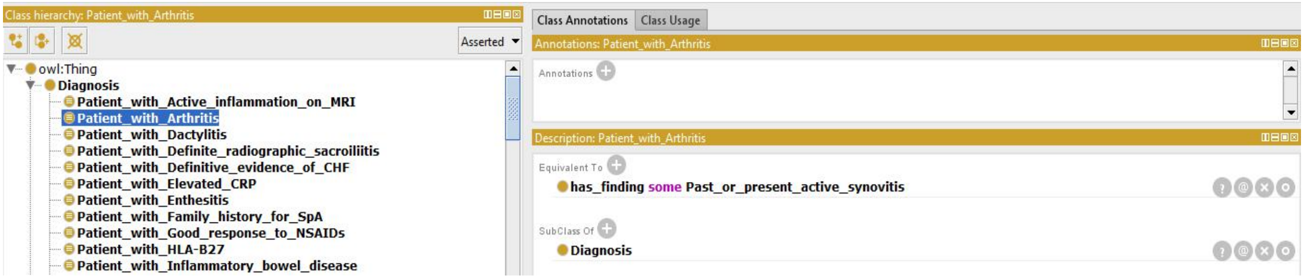

Figure 4.20: Description of diagnosis class *Arthritis* in OWL DL-based ontology.

Figure 4.21 displays a patient who has sign/symptom *Past_or_present_active_synovitis* as system input. Since Patient 1 has a set of clinical signs/symptoms, we use “and” to denote logical conjunction.

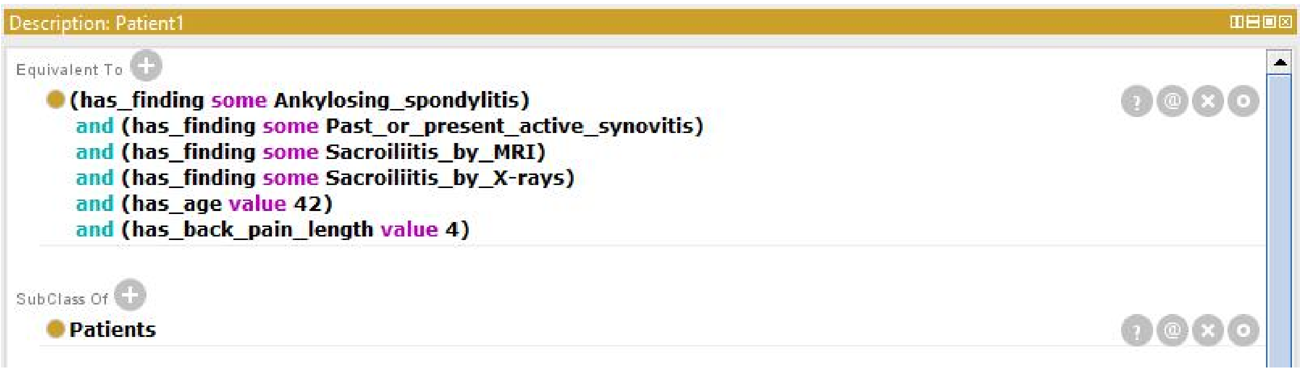

Figure 4.21: Descriptive knowledge of class *Patient1*.

For disease classes, we describe diseases SpA, IBP and DHF within the same model. For instance, disease SpA is described in Figure 4.22. Note that we use all object and data properties in this descriptive knowledge, and they align to produce disease identification for patients. In order to set numerical value constraints, we announce data type after each data property statement, such as *xsd: interger[]* and *xsd: float[]*. Disease IBP and DHF are expressed in a similar way.

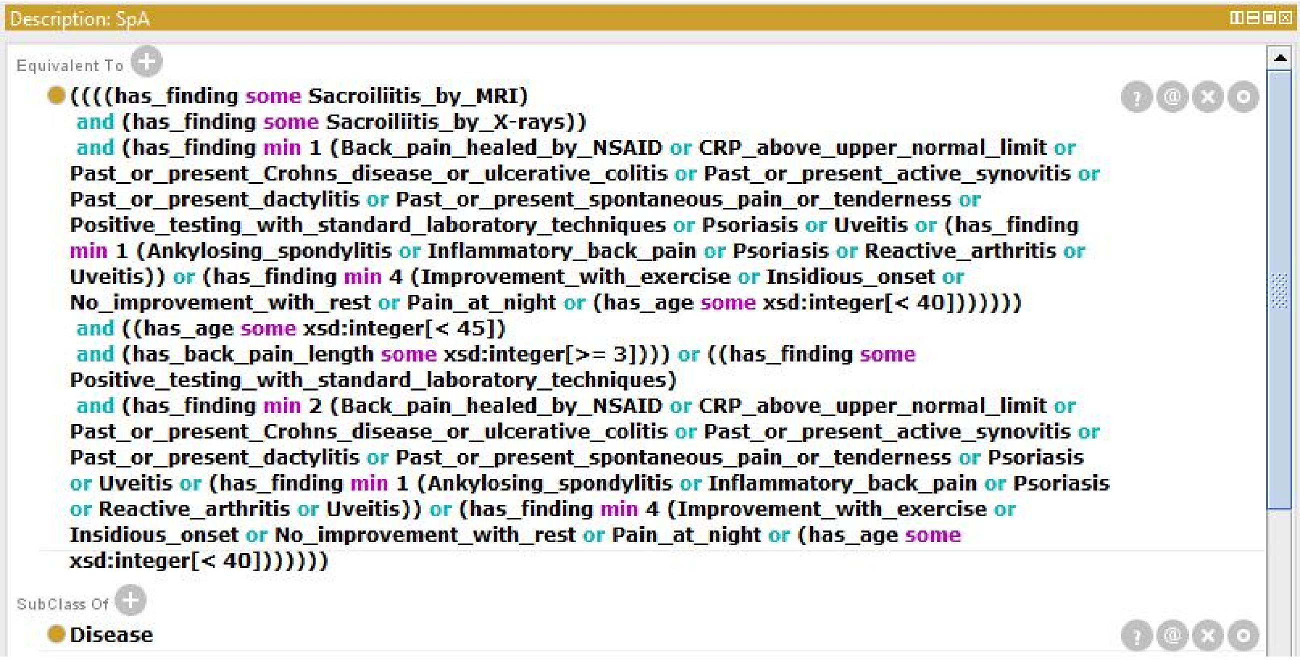

Figure 4.22: Descriptive knowledge of Disease class SpA.

In this way, we express all classes and subclasses with descriptive knowledge. In total, 59 classes are created with their description, to implement ontology reasoning via OWL DL.

4.2.3 Results Representation and Evaluation

After running the reasoner Hermit, results could be represented and stored. Figure 4.23 shows the results of Patient 1. We could observe that this patient is classified under a series of diagnosis classes and disease class *SpA*. In addition, since he is over 40 years old, and has back pain length for more than 3 months, he is also diagnosed to be possibly sickened by SpA.

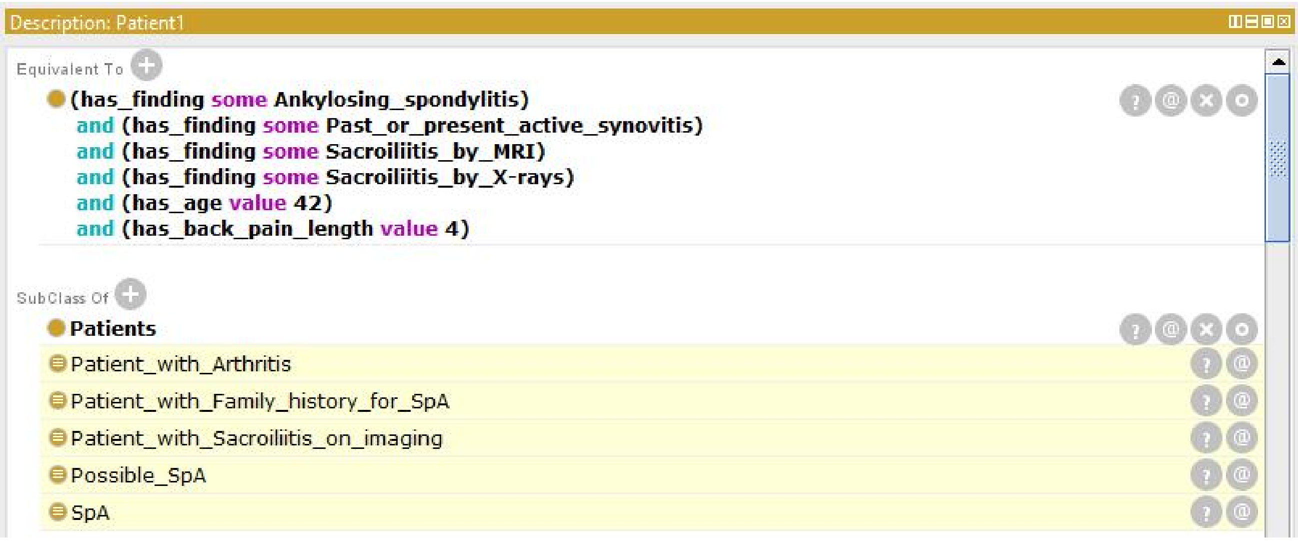

Figure 4.23: Results of Patient 1 in OWL DL-based ontology.

OWL DL also enables classes to obtain anonymous ancestors after ontology reasoning. In this part of descriptive knowledge, users could retrieve the information of subordinate relationships between classes, in the descriptive knowledge format. Figure 4.24 shows this aspect of results.

Figure 4.24: Subordinate relationships between Patient 1class and other classes.

Following this way, we obtain results representation of all 8 patients.

4.2.4 Results Evaluation

Results from OWL DL-based reanoning are also evaluated according to medical criteria that is defined by domain experts. In this subsection, we demonstrate one example which shows how diagnosis and identification of disease DHF works in the system.

We recall the knowledge introduced in Chapter 3.1. Disease DHF is diagnosed by the following statement:

((CHF_event_length some xsd:integer[<= 72]) and (measured_LV_EF_value some xsd:float[>= 0.5f])) and (presents some Objective_evidence_according_to_SNOMED_CT) and (presents some Abnormal_LV_relaxation/filling/distensibility_indices_on_cardiac_catheterization).

In the ontology, Patient 5 presents *signs/symptoms No_conclusive_information_on_LV_diastolic_function* and *Objective_evidence_according_to_SNOMED_CT*. His clinical input is as follows:
(has_finding some No_conclusive_information_on_LV_diastolic_function) and (has_finding some Objective_evidence_according_to_SNOMED_CT) and (CHF_event_length value 70) and (has_age value 36) and (has_back_pain_length value 7) and (measured_LV_EF_value value 0.6f).

The results are shown in Figure 4.24. We could observe that, since Patient 5 satisfies the requirements of clinical characteristics, he is classified under the disease DHF. The same mechanism is also successfully implemented on disease SpA.

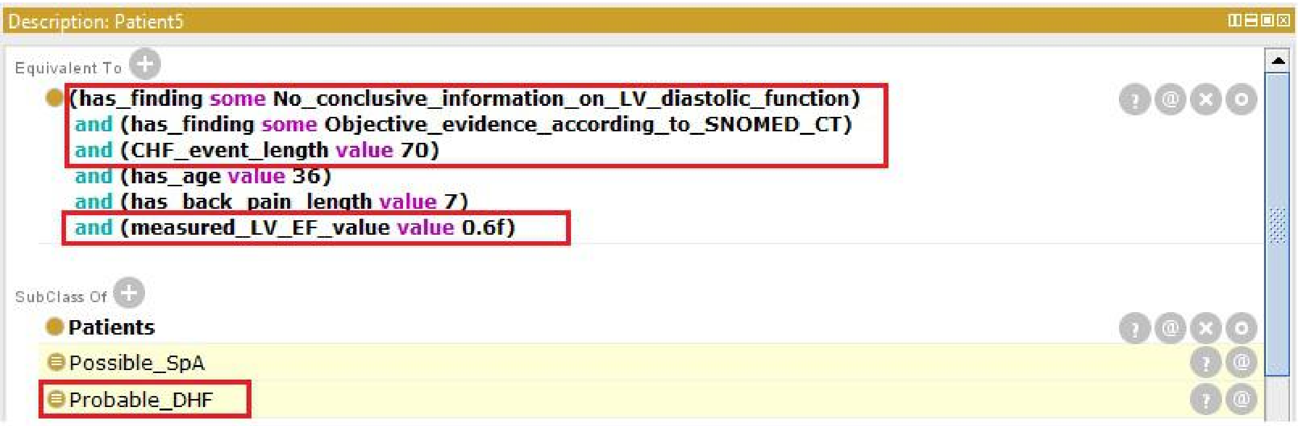

Figure 4.25: Results of diagnosis and disease identifications for Patient 5.

Following this methodology, all the results from OWL DL-based ontology reasoning are examined. The results show that the automatic diagnosis and disease identifications match with the knowledge that accepted by society.

4.3 Reasoning via SPARQL

Inspired by SQL, SPARQL is an RDF query language which is able to retrieve and manipulate stored data in RDF format, thus it enables us to access the knowledge from RDF/RDFS knowledge bases. Not only allowing users to extract and explore data via queries for unknown relations, it also has the capability of generating new RDF graphs based on already extent RDF query graphs. This functionality effectuates the construction of new knowledge. In other words, we could implement ontology reasoning via SPARQL, in order to derive hidden patterns in the created ontology.

To implement query and reasoning on ontology, Jena Fuseki is used to serve RDF data over H-TTP. Jena Fuseki is a SPARQL server which provides REST-style SPARQL update over HTTP and OWL files, and it also allows SPARQL query and update functions [50]. The step-by-step workflow is as follows:

1. Server installation. We first show how we got started with Jena Fuseki, by running the Fuseki server. After the server logging goes to the console, we then updated the ontology file which was created in Chapter 3.3, as an input dataset. Thereafter, we could apply SPARQL rules and queries to the ontology.
2. SPARQL rule construction. Similar to aforementioned two reasoning approaches, we constructed our rules in SPARQL. These rules were constructed via SPARQL update functionality, which means the rules were imported to the system as input, to derive and investigate hidden patterns and knowledge in the ontology.
3. Results retrieval and evaluation. Different from SWRL and OWL DL-based ontology reasoning results, output of SPARQL reasoning is not presented on the user interface. Instead, we need to use SPARQL query to retrieve the results we got. In consideration of this, we implemented several SPARQL queries, aligned with SPARQL update, to represent our results. We also checked the correctness and level of accuracy of the results.

4.3.1 Server Installation

The server we installed is Apache-jena-fuseki 2.4.1. After running the server, we reached the control panel, on which we could upload our ontology files as datasets. The web page offers SPARQL operations and data update on the selected dataset. Figure 4.26 shows how we reached the server page.

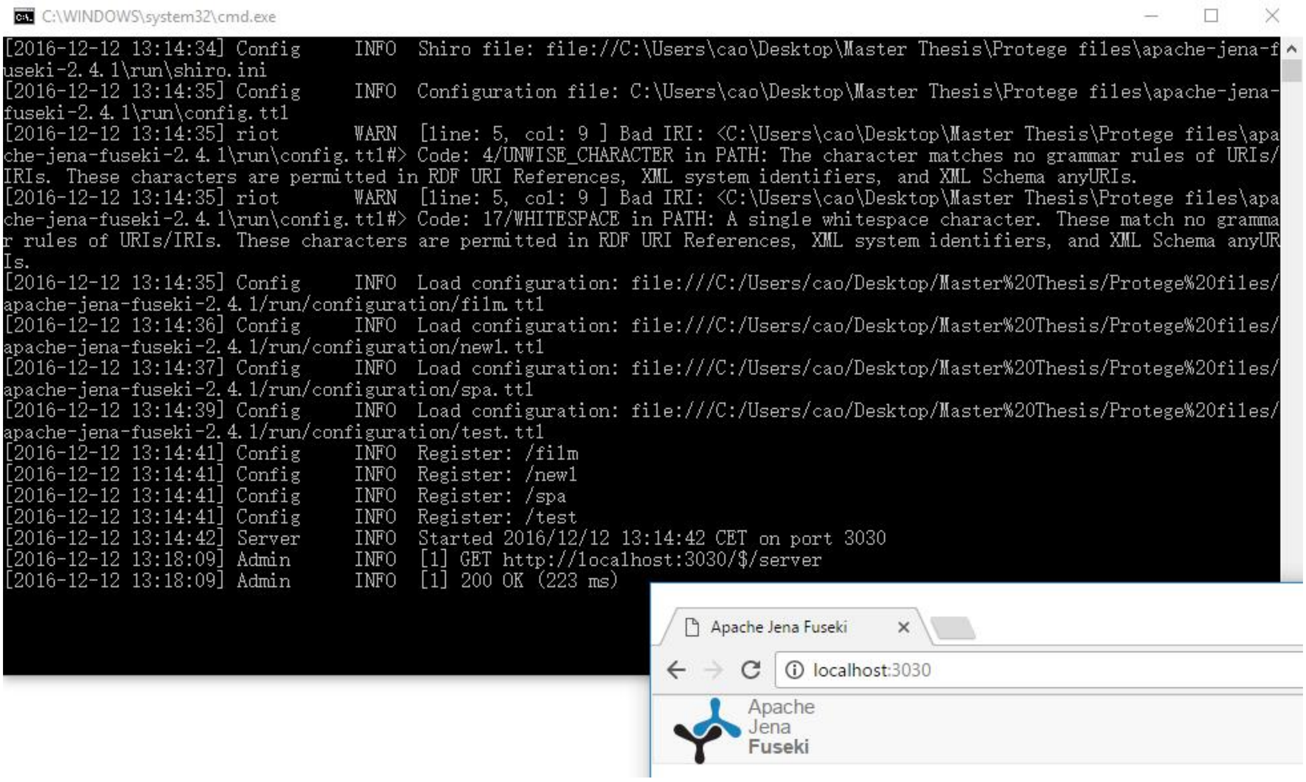

Figure 4.26: Screenshot about how Jena Fuseki serve runs.

After we run Fuseki server, we could manage datasets in Apache Jena Fuseki page. Here we uploaded the file we created in Chapter 3.3. The file is of .ttl type, and consists of all the triples we created. These triples include 23 classes, 62 individuals, 4 object properties and 6 data properties. Figure 4.27 shows the successful upload of the .ttl file. We can see the file contains information of 3006 triples.

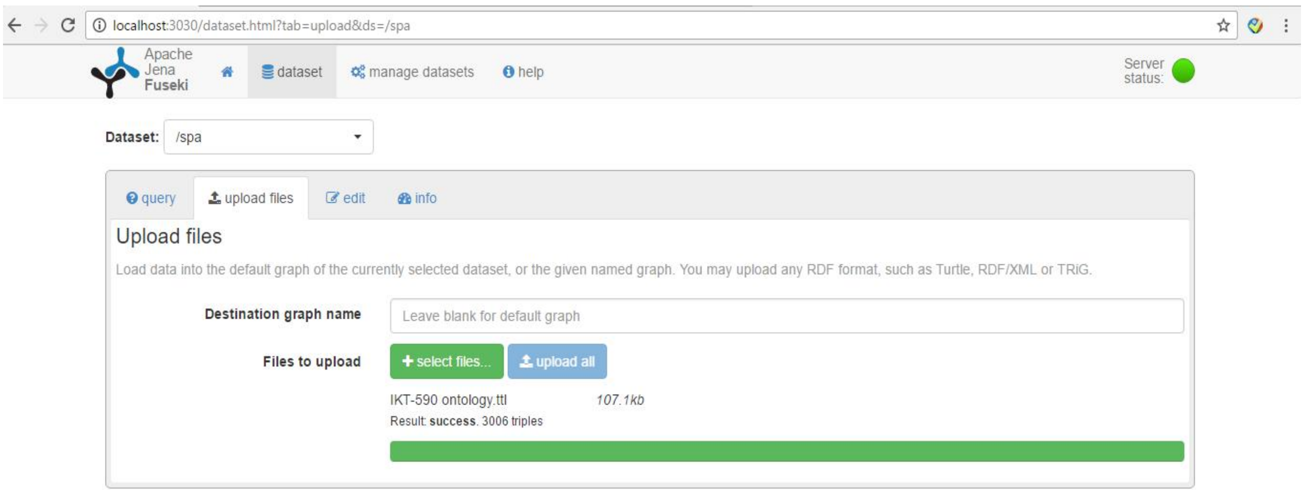

Figure 4.27: Successful upload of .ttl file in the dataset.

By this step, the Jena Fuseki server has been successfully installed.

4.3.2 SPARQL Rule Construction

As we discussed before, Jena Fuseki supports the service of SPARQL update. The update language for RDF graphs uses a syntax originated from SPARQL Query Language, and it performs on a collection of RDF graphs in graph store [51]. To execute update to RDF graphs, we use **Insert Data** operation to add triples into the ontology, as a method to implement ontology reasoning. According to W3C Recommendation, the format of insert function should be:

INSERT DATA QuadData,

where **QuadData** is formed by triple templates [52]. In this way, we constructed a set of SPARQL rules. In this section, we give one example among them, in which we will show how SPARQL rule construction worked.

For diagnosis Arthritis, the following SPARQL rule was constructed to execute diagnosis on patients.

SPARQL rule:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.semanticweb.org/cao/ontologies/2016/10/untitled-ontology-25#>
PREFIX swrl: <http://www.w3.org/2003/11/swrl#>
PREFIX swrla: <http://swrl.stanford.edu/ontologies/3.3/swrla.owl#>
PREFIX swrlb: <http://www.w3.org/2003/11/swrlb#>
INSERT   {?patient :has_diagnosis :Arthritis}
WHERE {
   ?patient a :Patient .
   ?patient :presents :Past_or_present_active_synovitis .
}
```

Note that there are a list of PREFIXS in the rule. The role of these PREFIXS are to enable serialization of Internationalized Resource Identifiers (IRI), and these IRIs aligned to simplify the names of IRIS in Turtle files. In this rule, we insert a triple in which the subject is Patient, object is Arthritis, and predication is has_diagnosis. Inside the triple, the subject Patient is defined by another triple, who has Past_or_present_active_synovitis as its object. As thus, one rule for ontology reasoning is constructed.

To see how this rule is inserted in the ontology, we implement SPARQL Update function, and results are shown in Figure 4.28. From query results window, we can see the rule is updated successfully in the dataset.

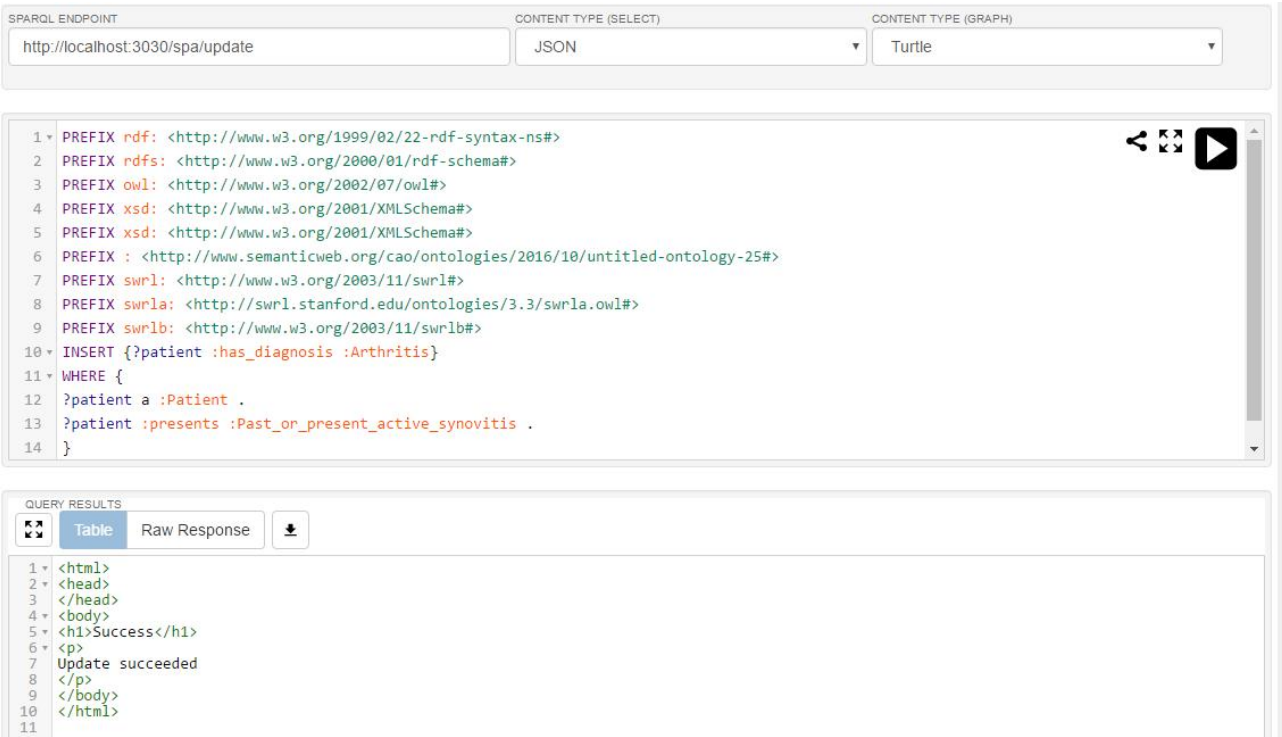

Figure 4.28: One example of SPARQL rule works in Jena Fuseki. Update to the dataset succeeded.

4.3.3 Results Retrieval and Evaluation

Different from SWRL and OWL DL ontology reasoning, the results from SPARQL rule reasoning would not show automatically in the server. Instead, we need to execute SPARQL query to retrieve our results. These retrieval statements are called SPARQL queries. To implement SPARQL queries, we should also write codes in RDF triples format, but instead of using **INSERT DATA** function, we use **SELECT** statement, which aims to find existing triple patterns in the ontology turtle file. Here we show the codes dealing with retrieving the results of SPARQL rule introduced in previous section.

SPARQL Query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.semanticweb.org/cao/ontologies/2016/10/untitled-ontology-25#>
PREFIX swrl: <http://www.w3.org/2003/11/swrl#>
PREFIX swrla: <http://swrl.stanford.edu/ontologies/3.3/swrla.owl#>
PREFIX swrlb: <http://www.w3.org/2003/11/swrlb#>
select ?patient ?diagnosis
WHERE {
?patient a :Patient .
?patient :has_diagnosis ?diagnosis .
}
```

The query correctly finds the corresponding triple in the ontology, proving our work of SPARQL rule construction is successful. The results are displayed in Figure 4.29.

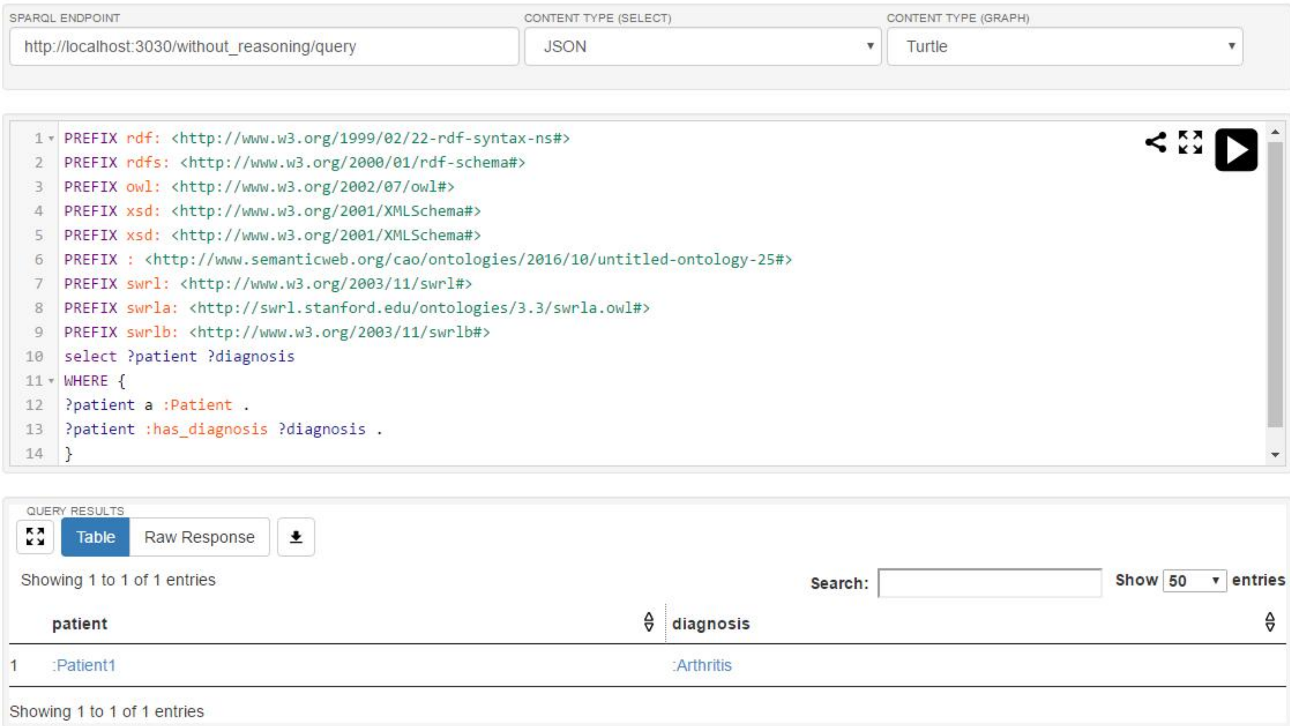

Figure 4.29: One example of SPARQL query worked in Jena Fuseki. Query to the dataset succeeded.

Following this approach, 14 rules were constructed and evaluated to produce diagnosis for patients. All the results were examined, and they proved to be feasible. Due to the time limit, we did not implement SPARQL rules on diseases level. This work could be a potential direction of future work.

4.4 Summary of the Chapter

Ontology reasoning is an approach that deals with decision support issues in many intelligent information systems. For the domains of biology and medicine, ontologies and associated generic tools could assist with the process of logical expression with no uncertainty. In this work, we explore three ontology reasoning technologies: SWRL which uses horn-like rules in IF-THEN logical format, OWL DL which is a formal logic-based knowledge representation language, and SPARQL which is a RDF query language, with capabilities of query and update in ontologies. All three reasoning approaches worked successfully, thereafter produced satisfactory results. All the outputted results were checked according to medical terminologies and terms which are defined by domain experts, to ensure that they accorded with the domain knowledge that are accepted by learned society.

Chapter 5: Results Demonstration and Discussion

In this chapter, the results of three ontology reasoning technologies are presented. We first demonstrate what we find in terms of advantages/disadvantages and similarities/differences after implementing SWRL, OWL DL and SPARQL reasoning approaches. Discussion upon results is to be carried out thereafter.

5.1 Results Demonstration

In Chapter 4, we introduced a novel approach to execute diagnosis and disease identifications on disease SpA, IBP and DHF. It would be legitimate to ask if our proposed methods are appropriate for achieving main goals of this thesis. As a consequence, in this section we demonstrate our results in a comparative manner, which means we evaluate the results in a parallel way, to test the efficacy of SWRL, OWL DL and SPARQL reasoning technologies. This part of effort aims to test the hypothesis 3 proposed in section 1.3.

5.1.1 SWRL vs OWL DL

Both SWRL reasoning and OWL DL reasoning were tested in Protégé. We select same patients to carry out the comparison, in order to see what similarities/differences and advantages/disadvantages we could find. To test efficacy of SWRL and OWL DL, we first demonstrate the results of Patient 1 in the ontology. Figure 5.1 shows a comparison between these two methods, in which the upper part contains the results from SWRL reasoning, and bottom part shows results we get from OWL DL reasoning.

From Figure 5.1 we observe both approaches correctly classify Patient 1 under class *Possible_SpA_Patient*, but SWRL provide diagnosis through property assertions of individuals, while OWL DL classifies Patient 1 by the use of subclass relationships. The reason behind this is, we created all diagnosis entities as individuals in SWRL ontology, but classes in OWL DL ontology. Thus, results are represented in different data storehouses (subclasses or individuals).

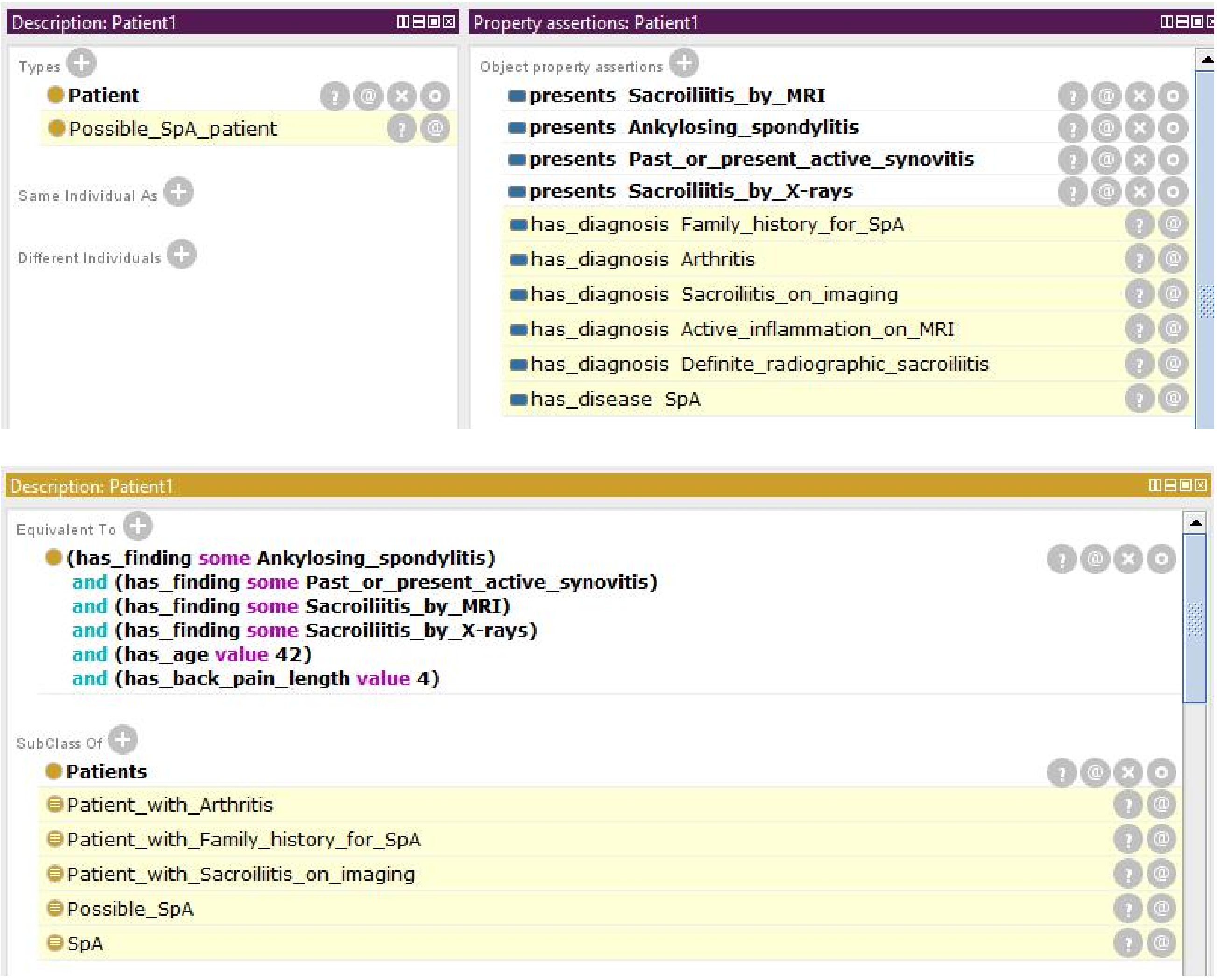

Figure 5.1: Comparison of results representation between SWRL and OWL DL. Above part are results from SWRL approach, and bottom part shows results of OWL DL approach.

Another notable difference is the number of diagnosis outputs. In SWRL reasoning, Patient 1 has 5 diagnosis, but only 3 in OWL DL. We could observe that diagnosis *Active_inflammation_on_MRI* and diagnosis *Definite_radiographic_sacroiliitis* are lacking in OWL DL-based ontology. In retrospect, sacroiliitis on imaging is defined by two diagnostic elements in ASAS [37]:

- Active inflammation on MRI
- Definite radiographic sacroiliitis.

Since these two pieces of diagnosis are represented with individuals in SWRL ontology, the diagnosis results could be presented as instances in property assertions. But for OWL DL-based ontology, these two diagnostic elements were represented as subclasses of superclass *diagnosis*. As a consequence, diagnosis *Patient_with_Sacroiliitis_on_imaging* contains these two diagnostic elements in OWL DL-based ontology. Only after we went deep into the result class *Patient_with_croiliitis_on_imaging*, two sub-diagnosis classes emerged. Figure 5.2 shows the representation in which sub-diagnosis *Active_inflammation_on_ MRI* and *Definite_radiographic_sacroilits* were unfolded.

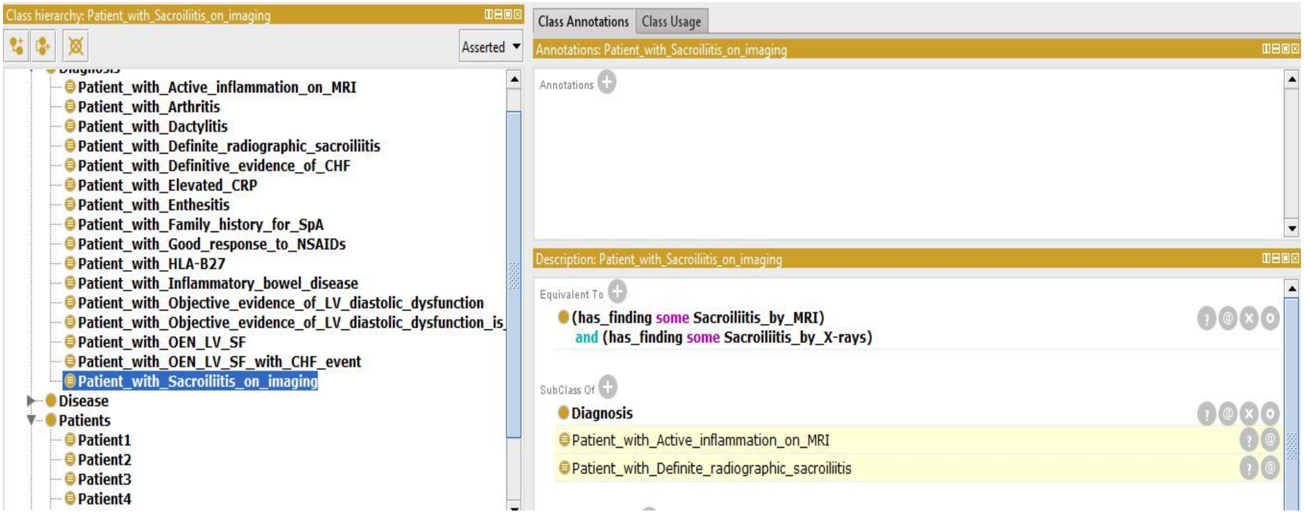

Figure 5.2: Results representation in which two diagnosis subclasses were unfolded.

Figure 5.1 and 5.2 combine to illustrate the issue that OWL DL reasoning happens on class level, which means the reasoning approach is parallel with class hierarchy. In contrast, in SWRL-based ontology reasoning, rules perform as intermediate data storage atoms. It means that for those rules which aim at individuals in ontologies, corresponding results could be stored in individuals elements under classes. Therefore, results are presented in a more complete manner than OWL DL-based reasoning.

We then illustrate this phenomena with a more completed example. Figure 5.3 and Figure 5.4 s-how how Patient 7 is diagnosed by the system. Having a list of signs/symptoms, Patient 7 is d-iagnosed to have *HLA-B27, Dyctylitis, OEN_LV_SF, Definite_evidence_of_C-HF, OEN_LV_SF_w-ith_CHF_event, Psoriasis* and *Objective_evidence_of_LV_diastolic_dysfunction_is_lacking* in SWRL based reasoning approach. Also, he is identified under diseases *IBP* and *SpA*. However, there is also a difference regarding to the number of diagnosis we got. While in O-WL DL-based reaso-ning, only diagnosis *Dactylitis* and *HLA-B27* were represented. To investigate the reason, we rec-all the knowledge from DHF classification schema and developed ontology. From Table 3.2-3.4, DHF is identified according to three levels of certainty: Definite, Probable and Possible, with d-owntrend possibility. This means that, Possible DHF is a sufficient condition of Probable DHF, and they two aligned to fulfill the sufficient conditions for Definite DHF.

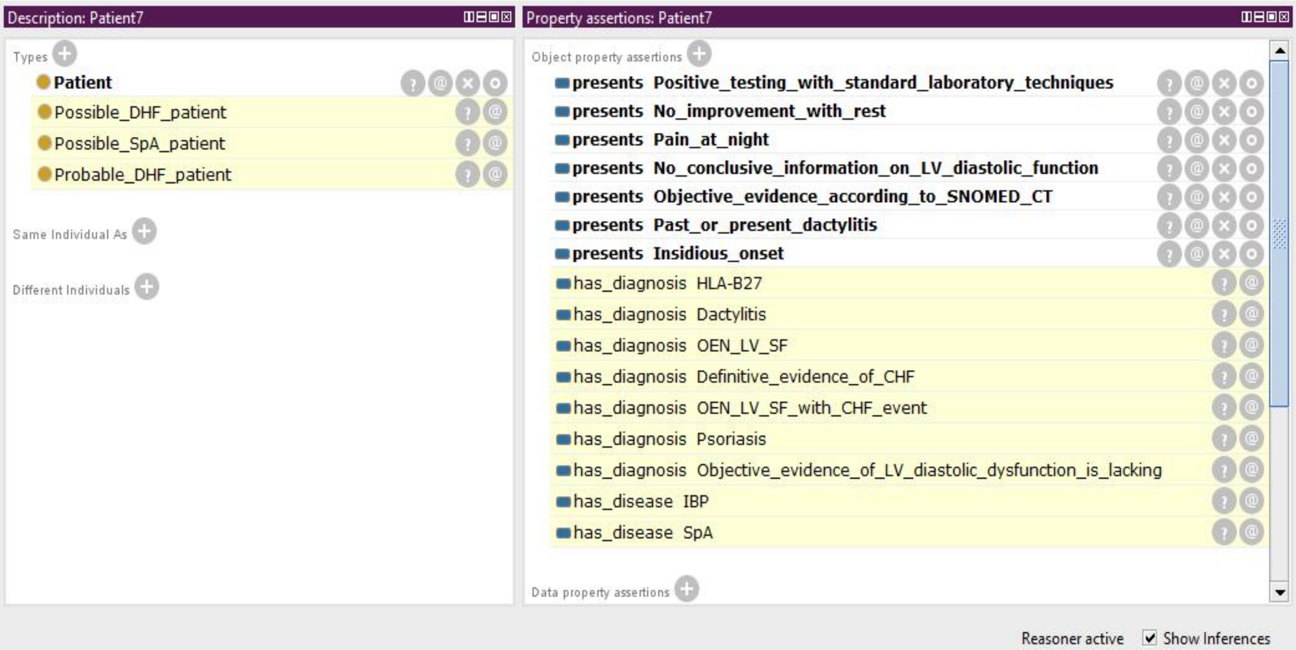

Figure 5.3: Results of Patient 7 in SWRL reasoning approach.

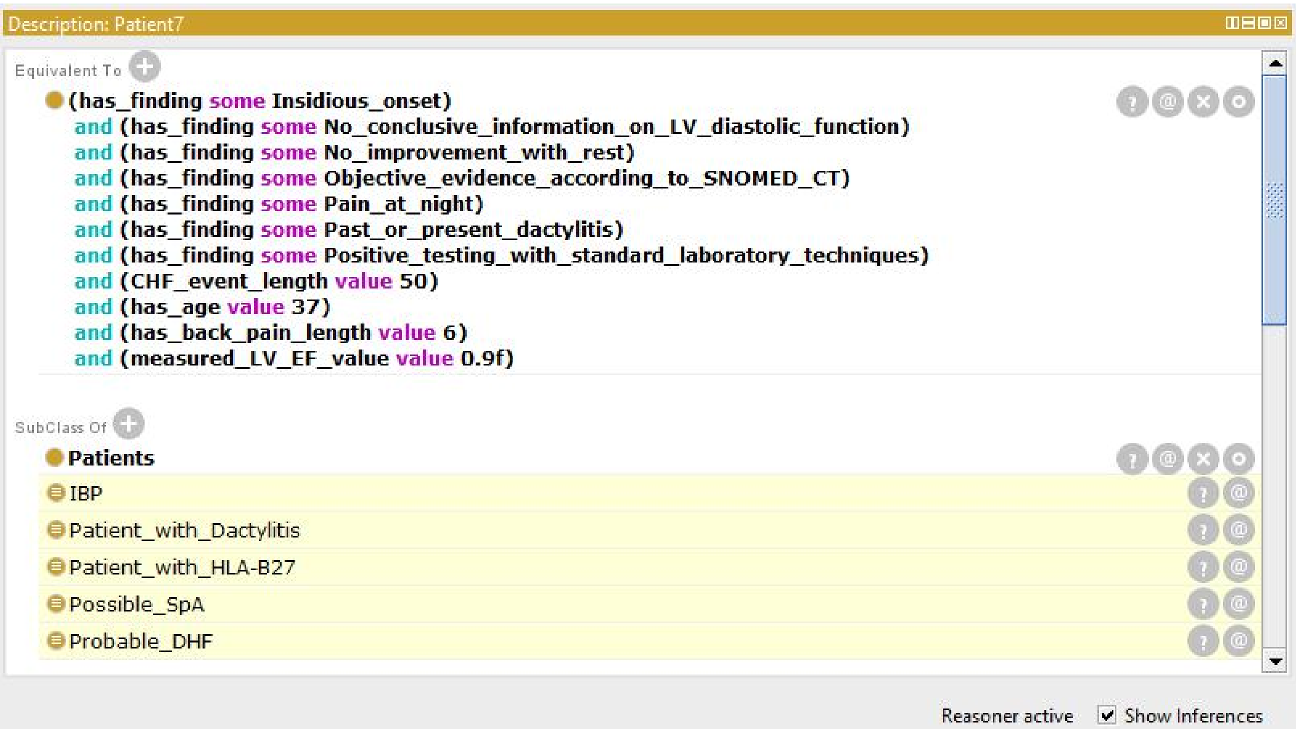

Figure 5.4: Results of Patient 7 in OWL DL-based reasoning approach.

We further explore the results representation in Figure 5.4. In descriptive knowledge of Patient 7, we click on the result class *Probable_DHF*. The resulting subclasses are shown in Figure 5.5. We could observe that diagnosis *Patient_with_OEN_LV_SF_with_CHF_event* springs forth, with another DHF identification subclass *Possible_DHF*.

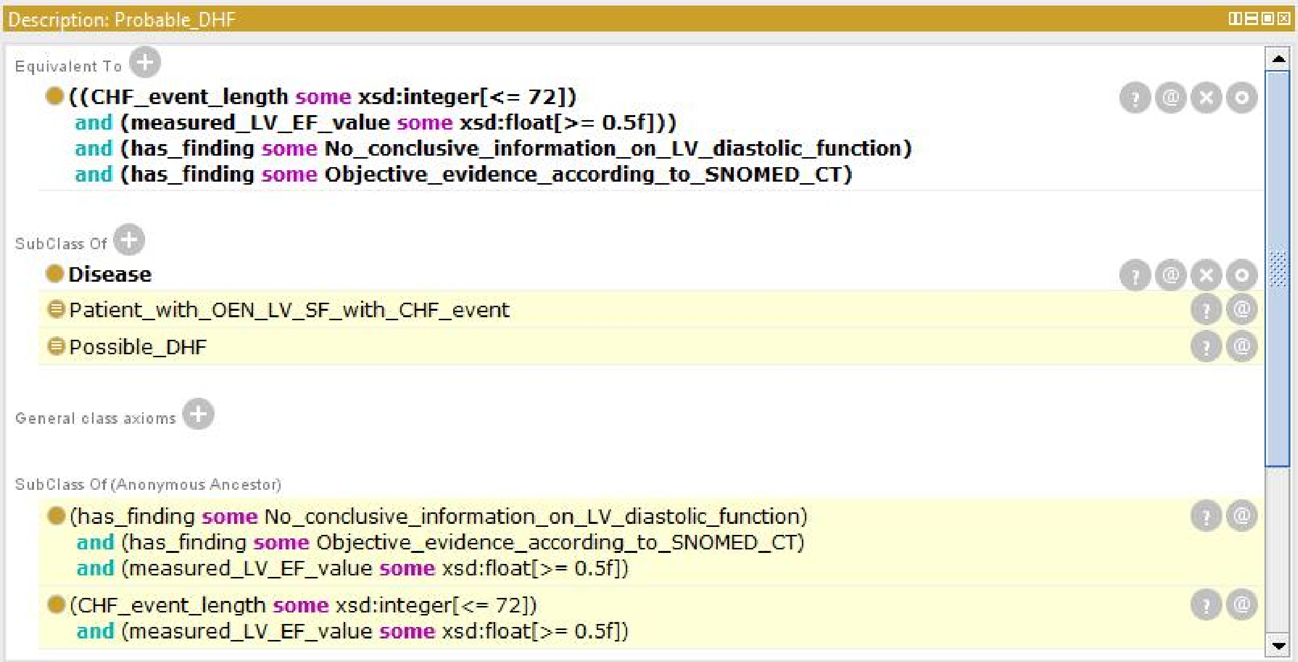

Figure 5.5: Descriptive knowledge of class Probable_DHF.

However, until this step we still have not obtained complete results. We then penetrate into subclass *Possible_DHF*, and Figure 5.6 shows what we achieve. Now we obtain all diagnosis results.

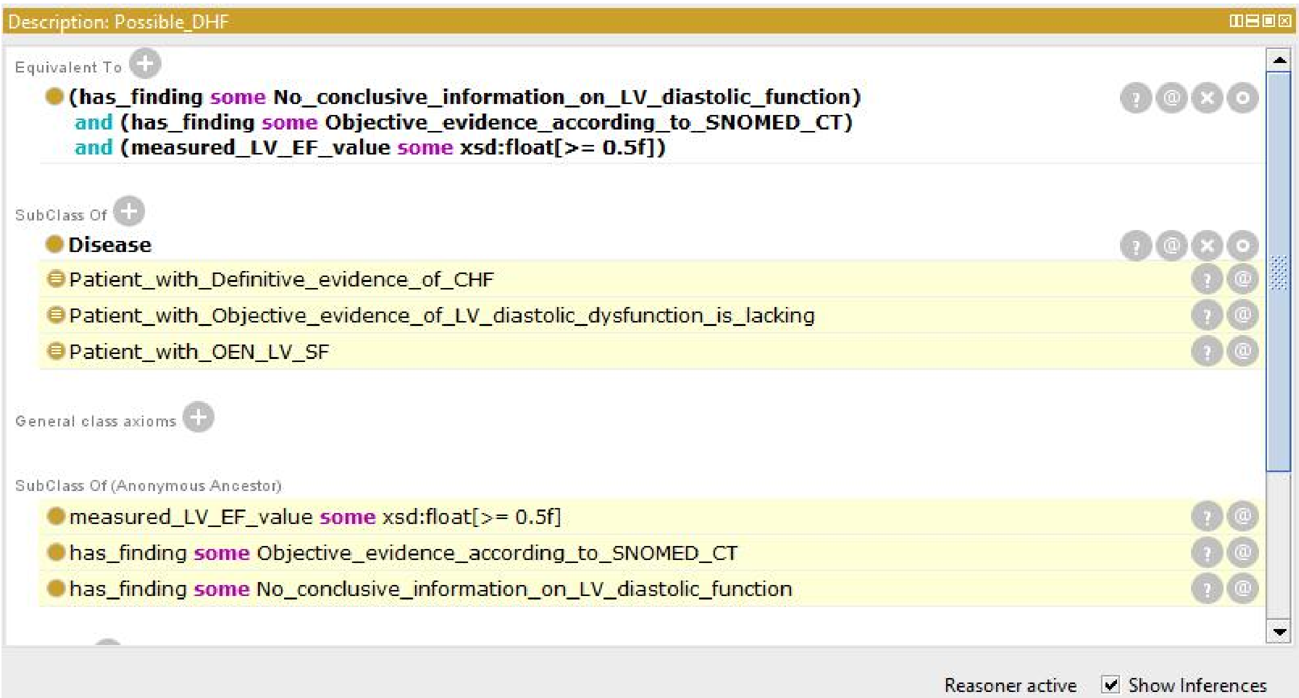

Figure 5.6: Descriptive knowledge of class Possible_DHF.

The differences between SWRL reasoning and OWL DL-based reasoning approaches are caused by distinct reasoning mechanisms. In SWRL reasoning, we reason on both individuals and classes in the ontology, while in OWL DL-based reasoning only classes and corresponding subclasses are involved. Table 5.1 shows the scale of both SWRL-based ontology and OWL DL-based ontology, in aspects of classes, individuals, data properties and object properties.

	Classes	**Individuals**	**Object properties**	**Data properties**
SWRL-based ontology	23	62	4	6
OWL DL-based ontology	59	0	4	6

Table 5.1: Scale of SWRL-based ontology and OWL DL-based ontology.

These two methods differ remarkably in terms of results representation when more individuals are created in the ontology. In this case, SWRL-based ontology reasoning has an advantage of representing results in a more comprehensive manner. When there exists subordinative relationships in ontologies, OWL DL-based reasoning was not brilliant enough to interpret all corresponding hidden patterns. In other words, it requires manual operations to completely demonstrate data of interest. One possible solution for this might be reasoning on both individuals and classes, instead of class-level DL reasoning.

Another issue we concern is the number of output elements we get. Table 5.2 shows the number of results pieces, where we find both two methods provided 51 pieces of information as output. It indicates that the results we obtained from SWRL reasoning and OWL DL-based reasoning match with each other.

	Number of outputs
SWRL-based reasoning	51
OWL DL-based reasoning	51

Table 5.2: Number of outputs from both reasoning approaches.

The results we obtained in this work could effectively prove that both SWRL and OWL DL are appropriate to implement ontology reasoning. Compared to the previous research such as [37], our CDSS succeeded to identify more diseases, and one extra reasoning approach based on SWRL was also introduced in the diagnosis and disease identification work.

5.2 Results Retrieval in SPARQL

In section 4.3, we constructed SPARQL rules and examined the rationality of them. In this section, we show how these rules aligned to produce outputs. SPARQL queries are used to check the results. The SPARQL queries we use are to perform a different role in this section. Instead of mining results coming from rules, these queries aim to extract data of interests from the whole ontology.

SPARQL rule 1: Diagnosis for Sacroiliitis on imaging.

This query is used to make diagnosis for patients who have *Sacroiliitis on imaging*. From section 3.1 we know that one patient who has *Sacroiliitis_by_MRI* or *Sacroiliitis_by_X-rays* as signs/symptoms may be identified as having diagnosis *Sacroiliitis on imaging*.

```
prefix :         <http://www.semanticweb.org/cao/ontologies/2016/10/untitled-ontology-25#>
prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix owl:     <http://www.w3.org/2002/07/owl#>
prefix xml:     <http://www.w3.org/XML/1998/namespace>
prefix swrlb: <http://www.w3.org/2003/11/swrlb#>
prefix swrl:    <http://www.w3.org/2003/11/swrl#>
prefix xsd:     <http://www.w3.org/2001/XMLSchema#>
prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#>
prefix swrla: <http://swrl.stanford.edu/ontologies/3.3/swrla.owl#>

INSERT { ?Patient :has_diagnosis :Sacroiliitis_on_imaging . }
WHERE {{?Patient :presents :Sacroiliitis_by_MRI.}
Union{
?Patient :presents :Sacroiliitis_by_X-rays.    }
}
```

SPARQL query 1:

select ?patient ?diagnosis

where {

?patient :has_diagnosis :Sacroiliitis_on_imaging.

?patient :has_diagnosis ?diagnosis

}

Figure 5.7 shows the results from query 1, where Patient 1, Patient 4, Patient 8 are extracted. This result matches with the ones in SWRL reasoning and OWL DL-based reasoning.

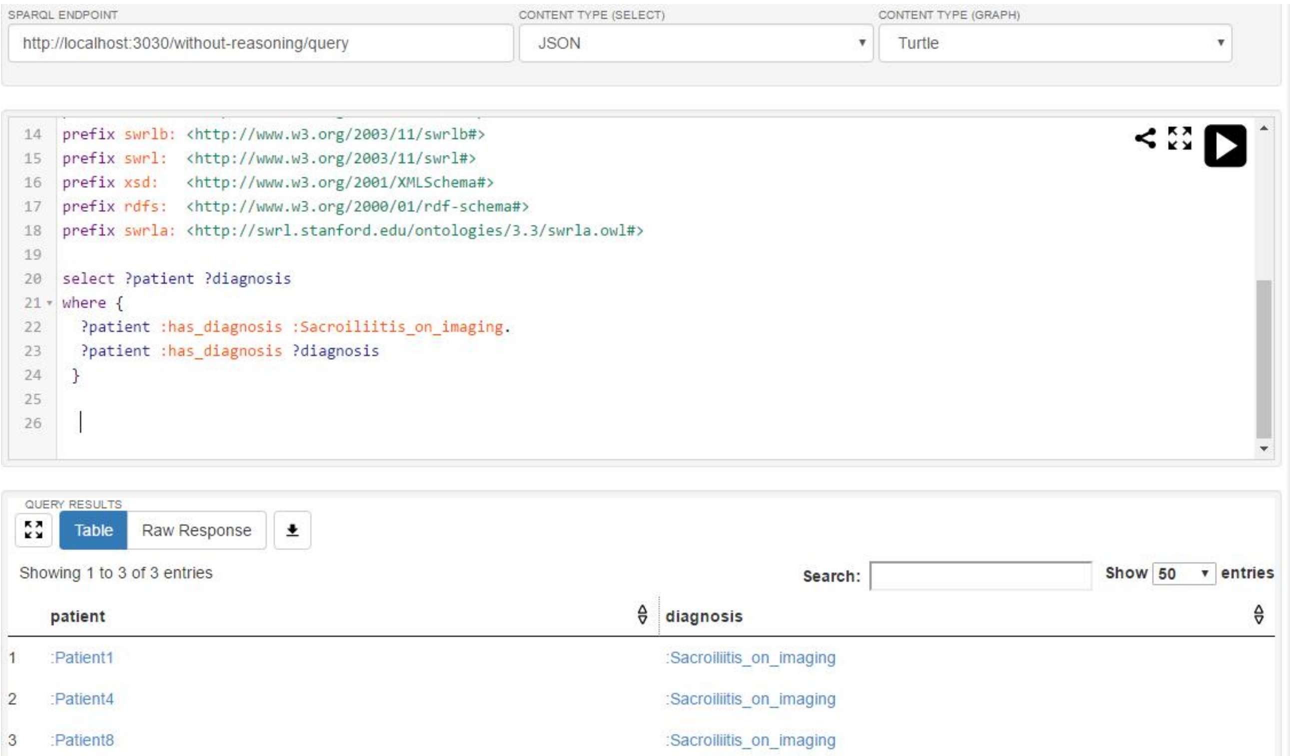

Figure 5.7: SPARQL query 1: Patients with diagnosis Sacroiliitis_on_imaging.

SPARQL rule 2: identify patients who are under risk of SpA.

In this rule, we aim to find patients who are possibly sickened with SpA. The similar principle is also used to look for patients with diseases IBP and DHF. Codes of rule 2 are as follows:

```
prefix :        <http://www.semanticweb.org/cao/ontologies/2016/10/untitled-ontology-25#>
prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix owl:     <http://www.w3.org/2002/07/owl#>
prefix xml:     <http://www.w3.org/XML/1998/namespace>
prefix swrlb: <http://www.w3.org/2003/11/swrlb#>
prefix swrl:    <http://www.w3.org/2003/11/swrl#>
prefix xsd:     <http://www.w3.org/2001/XMLSchema#>
prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#>
prefix swrla: <http://swrl.stanford.edu/ontologies/3.3/swrla.owl#>

INSERT { ?Patient :has_disease :Possible_SpA_patient }
WHERE {{?Patient :has_back_pain_length ?y.
    FILTER(?y > "3"^^xsd:integer)
        ?Patient :has_age ?x.
        Filter(?x < "45"^^xsd:integer)}    .
}
```

After we successfully inserted data, we implement corresponding SPARQL query to check our results.

SPARQL query 2:

```
Select ?Patient ?disease
where {?Patient :has_disease :Possible_SpA_patient}.
```

Results are shown in Figure 5.8. We see Patients 1, Patient 3, Patient 4, Patient 5, patient 7 and patient 8 are possible to have disease SpA. This result also matches with SWRL and OWL DL reasoning.

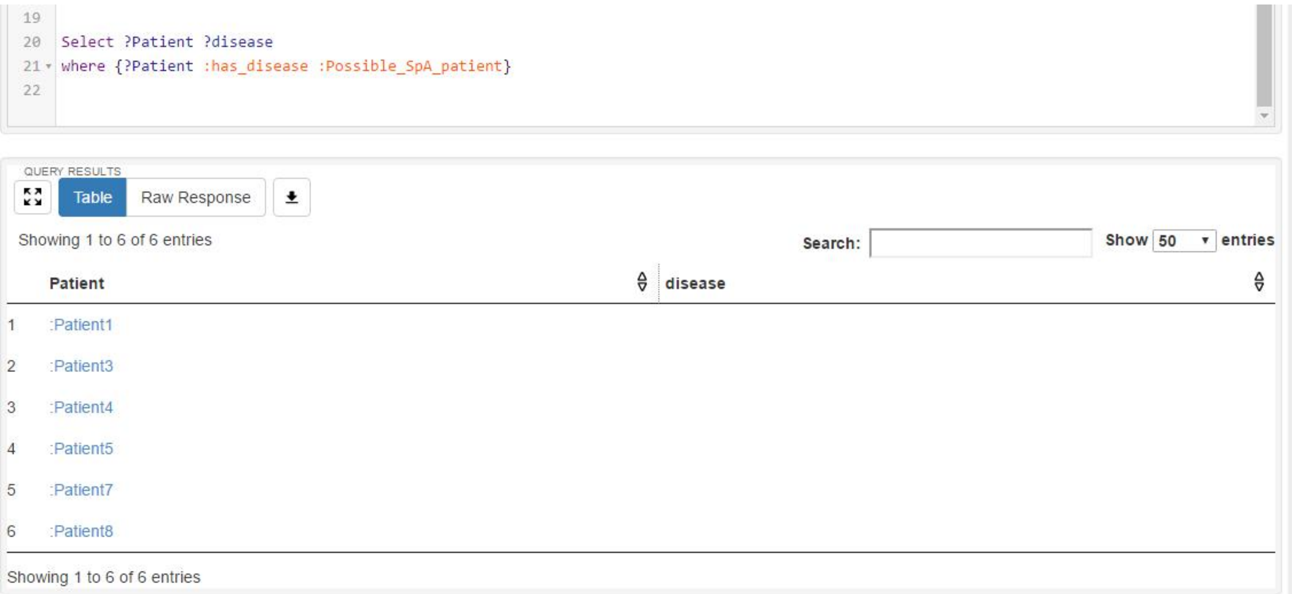

Figure 5.8: SPARQL query 2: Patients possibly have disease SpA.

SPARQL rule 3: disease IBP identification.

This rule aims to identify the disease IBP. Without using the object property *has_diagnosis,* triples with information about disease IBP are inserted. Codes are as follows:

prefix : <http://www.semanticweb.org/cao/ontologies/2016/10/untitled-ontology-25#>

prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

prefix owl: <http://www.w3.org/2002/07/owl#>

prefix xml: <http://www.w3.org/XML/1998/namespace>

prefix swrlb: <http://www.w3.org/2003/11/swrlb#>

prefix swrl: <http://www.w3.org/2003/11/swrl#>

prefix xsd: <http://www.w3.org/2001/XMLSchema#>

prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>

prefix swrla: <http://swrl.stanford.edu/ontologies/3.3/swrla.owl#>

INSERT { ?Patient :has_disease :IBP }

WHERE {{?Patient :has_age ?x.

Filter(?x < "40"^^xsd:integer)

?Patient :presents :Improvement_with_exercise.

?Patient :presents :No_improvement_with_rest.

```
        ?Patient :presents :Pain_at_night.}
  Union{?Patient :has_age ?x.
        Filter(?x < "40"^^xsd:integer)
        ?Patient :presents :Insidious_onset.
        ?Patient :presents :No_improvement_with_rest.
        ?Patient :presents :Pain_at_night.}
  Union{?Patient :has_age ?x.
        Filter(?x < "40"^^xsd:integer)
        ?Patient :presents :Insidious_onset.
        ?Patient :presents :Improvement_with_exercise.
        ?Patient :presents :Pain_at_night.}
  Union{?Patient :has_age ?x.
        Filter(?x < "40"^^xsd:integer)
        ?Patient :presents :Insidious_onset.
        ?Patient :presents :Improvement_with_exercise.
        ?Patient :presents :No_improvement_with_rest.}
  Union{?Patient:presents :No_improvement_with_rest.
        ?Patient :presents :Insidious_onset.
        ?Patient :presents :Improvement_with_exercise.
        ?Patient :presents :Pain_at_night.}
}
```

SPARQL query 3:

```
Select ?Paitient ?disease
where
{ ?Paitient :has_disease :IBP}
```

Figure 5.9 shows the results. Patient 2, Patient 4 and Patient 7 are classified under the disease IBP. We again check the correctness of the results, and they match with those we got from the other two ontology reasoning approaches.

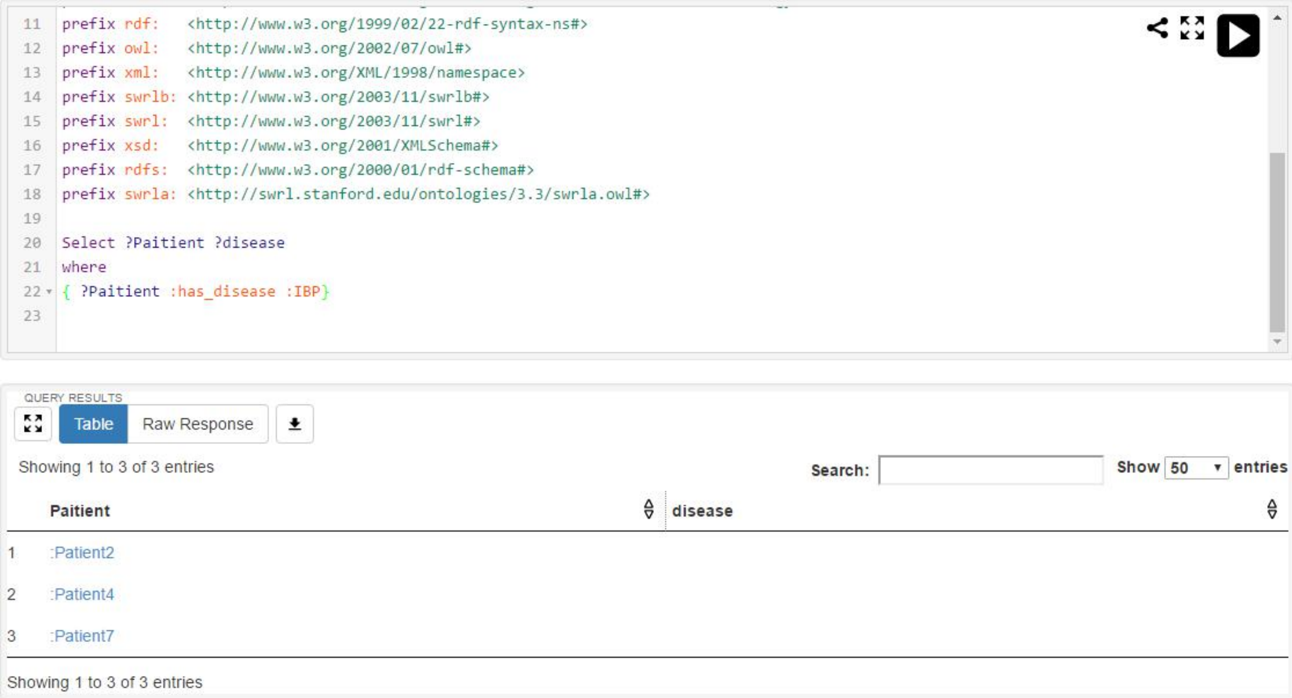

Figure 5.9: Results from SPARQL query 3: disease IBP identification.

In this way, we implemented 14 rules and 10 queries to check the results of SPARQL reasoning. The advantage of SPARQL in this study we find is that it enables information retrieval with a full set of analytic rules and query operations, and by using these operations we can easily insert, construct and access the data we are interested in. Different from SWRL and OWL DL which were implemented in Protégé, we did not need to navigate in the software user interface to find the data we want. In SPARQL, only one query might be needed to obtain all information of an atom in the ontology. It could significantly save our time in terms of manual operations.

5.3 Summary of the Chapter

In this chapter, the results of three ontology reasoning technologies were presented. We compared the efficacy, scale, advantages/disadvantages and similarities/differences among SWRL, OWL DL and SPARQL reasoning approaches. As a result, we found SWRL was more flexible than OWL DL, since Horn-like rules could perform as intermediate data storage entities in the ontology. Also, these rules are easier to maintain, since we could take operations like delete, edit, create new and comment on rules. These modification operations could be carried out without changing the structure of the ontology and

CDSS. For reasoning via SPARQL, the work was executed through Jena Fuseki server. It provides SPARQL query, SPARQL update functions, which enable the easy accessibility of data. This mechanism could significantly boost the efficiency of data retrieval work. In addition, we could tailor the queries to domain experts' need, by presenting data of their interests.

Chapter 6: Conclusion

In this work, we have designed medical ontologies and CDSS for diagnosis and disease identification. Based on others' work, we made our own contributions to propose an extended ontological representation to classify patients under diseases SpA, IBP and DHF. Particularly the development work introduced in section 3.1 was introduced to verify our hypothesis 1. By executing specification and formalization of existing medical terminologies and diagnostic criteria, we proved that advanced CDSS could be developed through a top-down approach.

To evaluate the veracity of hypothesis 2, we implement 3 ontology reasoning approaches: SWRL, OWL DL and SPARQL. Although the structure of ontologies used in these reasoning approaches were different, results showed that we could get the same results from all of them. In SWRL, 84 rules were constructed to reason on classes, subclasses and individuals in the ontology containing all diagnosis and signs/symptoms of disease SpA, IBP and DHF. According to our knowledge, this work has not been studied by others. In OWL DL-based reasoning approach, we extended others' work by introducing more diagnosis and disease atoms, and in total 59 classes were created to carry out the diagnosis and disease identification process. All descriptive knowledge was represented with Manchester OWL Syntax. While in SPARQL reasoning, we constructed 14 SPARQL rules and 10 SPARQL queries to proceed the reasoning process on diagnosis. However, more work is needed to include disease classes in the inference engine, to show the performance of SPARQL rules on classification work of patients.

The successful verification of first two hypothesises leads to the demonstration of the comparison work introduced in Chapter 5. We investigated advantages/disadvantages and similarities/differences among these three reasoning technologies, and drew conclusion that all of them are appropriate methods to carry out ontology reasoning. Also, interoperability of semantic web-based ontologies and associated generic tools are suitable to represent diagnosis and disease definitions. In this sense, we can identify diagnostic decisions and diseases with the help of semantic web-based CDSS.

Chapter 7: Future work

In this chapter, we suggest possible improvements to the proposed medical ontologies and several interesting directions of future work.

The work introduced in this thesis is currently an essential model of CDSS, which means it is not yet complete enough for clinical professionals and experts to use. To elevate the model to a usable CDSS, it would be interesting to develop ontology-based CDSS prototypes based on those ontologies described in this thesis. In this sight, evaluation and maintenance are required to test the acceptance of recommendations, satisfaction with the system and usefulness of the ontologies [53]. Interaction design methodologies including user study, usability test and user interface development are necessary to implement during the process of developing a CDSS.

The ontologies used in this work could be reused in other domains. Since one of the advantages of ontology technology is interoperability, it would be interesting to apply knowledge bases in this thesis to other diseases classification, like diabetes, cancer, joint pathology and bone pathology [9, 54, 55]. Also, these knowledge bases could also help with data set generation, hypothesis validation and hidden patterns discovery.

There are also several improvements that could be made to reasoning approaches. To effectively meet the requirements of context reasoning and modeling, further work about combining ontologies and rules are required. In order to build up more advanced ontology-based data analyze frameworks, we need to create and combine different ontologies in a specific domain, and construct more rules to cope with more complicated issues. In the reasoning approach with SPARQL, we used insert data functions to implement reasoning process. However, it is not strong enough to deal with all the aspects of ontology reasoning. There are other robust SPARQL reasoning methods such as SPIN vocabulary [59], to execute the SPARQL rules. The future work on this direction would be a valuable attempt on generating SPARQL-based activity recognition rules.

The patients data used in this work was manually imported to the ontologies. Under the environment of hospitals, it may be time consuming for clinicians to input data to the system manually. For the patients data that is stored in the patient record, it would significantly increase efficiency if machines could automatically extract and translate patients' textual record to machine-readable data. In the future, we plan to investigate how data science technologies like data retrieval and text mining could help with this issue.

References

[1] "Ontologies - introduction to ontologies and semantic web - tutorial,". [Online]. Available: http://obitko.com/tutorials/ontologies-semantic-web/ontologies.html. Accessed: Oct. 01, 2016.

[2] F. Freitas, S. Schulz, and E. Moraes, "Survey of current terminologies and ontologies in biology and medicine," *RECIIS*, vol. 3, no. 1, Mar. 2009.

[3] "Decision support system," in Wikipedia, Wikimedia Foundation, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Decision_support_system. Accessed: Sep. 11, 2016.

[4] M. V. Sokolova and A. Fernández Caballero, "A review on Frameworks for decision support systems," *Intelligent Systems Reference Library*. Springer Nature, 2012, pp. 19–45.

[5] "Clinical decision support system," in Wikipedia, Wikimedia Foundation, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Clinical_decision_support_system. Accessed: Sep. 21, 2016.

[6] A. Garg, N. Adhikari, H. McDonald, M. Rosas-Arellano, P. Devereaux, J. Beyene, J. Sam and R. Haynes, "Effects of Computerized Clinical Decision Support Systems on Practitioner Performance and Patient Outcomes", *JAMA*, vol. 293, no. 10, p. 1223, 2005.

[7] M.-M. Bouamrane, A. Rector, and M. Hurrell, "Using OWL ontologies for adaptive patient information modelling and preoperative clinical decision support," *Knowledge and Information Systems*, vol. 29, no. 2, pp. 405–418, Oct. 2010.

[8] K. Farion, W. Michalowski, S. Wilk, D. O´Sullivan, S. Rubin, and D. Weiss, "Clinical decision support system for point of care use - Ontology-driven design and software implementation," *Methods of Information in Medicine,* vol. 48, no. 4, pp. 381–390, May 2009.

[9] V. Bertaud-Gounot, R. Duvauferrier, and A. Burgun, "Ontology and medical diagnosis," *Informatics for Health and Social Care*, vol. 37, no. 2, pp. 51–61, Oct. 2011.

[10] M. Abbasi, S. Kashiyarndi, ``Clinical Decision Support Systems: A discussion on different methodologies used in Health Care,'' *Marlaedalen University*, Sweden, 2006.

[11] R. S. Ledley and L. B. Lusted, "Reasoning foundations of medical diagnosis: Symbolic logic, probability, and value theory aid our understanding of how physicians reason," *Science*, vol. 130, no. 3366, pp. 9–21, Jul. 1959.

[12] B. K. Philip, P. R. Reese, and S. P. Burch, "The economic impact of opioids on postoperative pain management," *Journal of Clinical Anesthesia*, vol. 14, no. 5, pp. 354–364, Aug. 2002.

[13] A. Wright and D. F. Sittig, "A four-phase model of the evolution of clinical decision support architectures," *International Journal of Medical Informatics,* vol. 77, no. 10, pp. 641–649, Oct. 2008.

[14] J. A. Stockman, "Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: A systematic review," *Yearbook of Pediatrics*, vol. 2006, pp. 293–295, Jan. 2006.

[15] M. V. Sokolova and A. Fernández Caballero, "A review on Frameworks for decision support systems," *Intelligent Systems Reference Library*. Springer Nature, 2012, pp. 19–45.

[16] J. Yang, U. Kang, and Y. Lee, "Clinical decision support system in medical knowledge literature review," *Information Technology and Management*, vol. 17, no. 1, pp. 5–14, Feb. 2015.

[17] J. Warren, G. Beliakov and B. van der Zwaag, "Fuzzy logic in clinical practice decision support systems," *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 2000, pp. 10.

[18] J. G. Anderson, "Clearing the way for physicians' use of clinical information systems," *Communications of the ACM*, vol. 40, no. 8, pp. 83–90, Jan. 1997.

[19] L. Zhang, H. Wang, X. Cao, X. Wang, and K. Zhao, "Knowledge management component in managing human resources for enterprises," *Information Technology and Management*, vol. 13, no. 4, pp. 341–349, May 2012.

[20] E. Mun Lim et al., "A discussion of cases in the 2001 RCPA-AQAP chemical pathology case report comments program," *Pathology*, vol. 35, no. 2, pp. 145–150, Apr. 2003.

[21] H. Liu and G. Hua, "Introduction to the special issue on information technologies in logistics and service science," *Information Technology and Management*, vol. 14, no. 3, pp. 167–168, Jun. 2013.

[22] G. O. Barnett, R. Winickoff, J. L. Dorsey, M. M. Morgan, and R. S. Lurie, "Quality assurance through automated monitoring and concurrent feedback using a computer-based medical information system," *Medical Care*, vol. 16, no. 11, pp. 962–970, Nov. 1978.

[23] M. Peleg, "Computer-interpretable clinical guidelines: A methodological review," *Journal of Biomedical Informatics*, vol. 46, no. 4, pp. 744–763, Aug. 2013.

[24] A. M. Fairchild, "Decision management: Role and effect of using an intelligent intermediary to aid in information sharing," *Information Technology and Management,* vol. 7, no. 4, pp. 249–258, Dec. 2006.

[25] R. N. Kostoff, "Database tomography for technical intelligence," *Competitive Intelligence Review*,

vol. 4, no. 1, pp. 38–43, 1993.

[26] F. Freitas, S. Schulz, and E. Moraes, "Survey of current terminologies and ontologies in biology and medicine," RECIIS, vol. 3, no. 1, Mar. 2009.

[27] D. L. Rubin, N. H. Shah, and N. F. Noy, "Biomedical ontologies: A functional perspective," *Briefings in Bioinformatics*, vol. 9, no. 1, pp. 75–90, Oct. 2007.

[28] H. Wasserman and J. Wang, "An applied evaluation of SNOMED CT as a clinical vocabulary for the computerized diagnosis and problem list," vol. 2003, 2003.

[29] P. L. Elkin et al., "Evaluation of the content coverage of SNOMED CT: Ability of SNOMED clinical terms to represent clinical problem lists," *Mayo Clinic Proceedings*, vol. 81, no. 6, pp. 741–748, Jun. 2006.

[30] G. Kong, D.-L. Xu, and J.-B. Yang, "Clinical decision support systems: A review on knowledge representation and inference under uncertainties," *International Journal of Computational Intelligence Systems*, vol. 1, no. 2, pp. 159–167, May 2008.

[31] J. Warren, G. Beliakov and B. van der Zwaag, "Fuzzy logic in clinical practice decision support systems," *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 2000, pp. 10.

[32] J. G. Anderson, "Clearing the way for physicians' use of clinical information systems," *Communications of the ACM*, vol. 40, no. 8, pp. 83–90, Jan. 1997.

[33] "Ontology," in Wikipedia, Wikimedia Foundation, 2016. [Online].
Available: https://en.wikipedia.org/wiki/Ontology.Accessed: Nov. 05, 2016.

[34] Stanford.edu. [Online].
Available:vhttp://Protégé.stanford.edu/publications/ontology_development/ontology101-noy-mcguin-ness.html].Accessed: Nov. 05, 2016.

[35] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific American, May 2001, pp. 28- 37.

[36] Wikimedia.org. [Online].
Available:https://upload.wikimedia.org/wikipedia/commons/f/f7/Semantic_web_stack.svg.Accessed: Nov. 16, 2016.

[37] V. Bertaud-Gounot, R. Duvauferrier, and A. Burgun, "Ontology and medical diagnosis," Informatics for Health and Social Care, vol. 37, no. 2, pp. 51–61, Oct. 2011.

[38] "Web Ontology language," in Wikipedia, Wikimedia Foundation, 2016. [Online].

Available: https://en.wikipedia.org/wiki/Web_Ontology_Language. Accessed: Dec. 16, 2016.

[39] "Protégé," 2016. [Online].

Available: http://Protégé.stanford.edu/.Accessed: Dec. 16, 2016.

[40] "Semantic reasoner," in Wikipedia, Wikimedia Foundation, 2016. [Online].

Available: https://en.wikipedia.org/wiki/Semantic_reasoner.Accessed: Nov. 16, 2016.

[41] "W3.org," [Online].

Available: https://www.w3.org/TR/rdf-sparql-query/.Accessed: Nov. 25, 2016.

[42] R. Koper, "Use of the semantic web to solve some basic problems in education: Increase flexible, distributed lifelong learning; decrease teacher's workload," *Journal of Interactive Media in Education*, vol. 2004, no. 1, p. 5, May 2004.

[43] T.-E. Hokstad, "Ontology based study planning and classification of university subjects," Universitetet i Agder ; University of Agder, 2015.

[44] I. Horrocks et al., "SWRL: A semantic web rule language combining OWL and RuleML," 2004. [Online].

[45] J. P. Nytun, "Reasoning with rules SWRL as example," 2016. [Online].

Available:http://grimstad.uia.no/janpn/IKT437/2016/slides/pdf/rules_2016.pdf. Accessed: Dec. 01, 2016.

[46] A. Jovic, M. Prcela and D. Gamberger, "Ontologies in Medical Knowledge Representation," 2007 29th International Conference on Information Technology Interfaces, Cavtat, 2007, pp. 535-540.

[47] A. Jović, D. Gamberger, and G. Krstačić, "Croatian scientific bibliography browsing paper," *Bio-Algorithms and Med-Systems*, vol. 7, no. 2, pp. 101–110, Jan. 2011.

[48] R. S. Vasan and D. Levy, "Defining Diastolic heart failure: A call for standardized diagnostic criteria," *Circulation,* vol. 101, no. 17, pp. 2118-2121, May 2000.

[49] C. Zanni-Merk, "KREM: A generic knowledge-based framework for problem solving in engineering - proposal and case studies," Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 2015.

[50] "Fuseki: Serving RDF data over HTTP," 2011. [Online].

Available: https://jena.apache.org/documentation/serving_data/. Accessed: Dec. 01, 2016.

[51] "SPARQL 1.1 update," 2013. [Online].

Available: https://www.w3.org/TR/sparql11-update/. Accessed: Dec. 05, 2016.

[52] "SPARQL 1.1 update," 2013. [Online].

Available: https://www.w3.org/TR/2013/REC-sparql11-update-20130321/#insertData.

Accessed: Dec. 11, 2016.

[54] G. Meditskos, S. Dasiopoulou, V. Efstathiou and I. Kompatsiaris, "SP-ACT: A hybrid framework for complex activity recognition combining OWL and SPARQL rules," 2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), San Diego, CA, 2013, pp. 25-30.

[55] C. Buil-Aranda, A. Hogan, J. Umbrich, and P.-Y. Vandenbussche, "SPARQL web-querying infrastructure: Ready for action?," *Lecture Notes in Computer Science*. Springer Nature, 2013, pp. 277–293.