

Semantic Web System for Differential Diagnosis Recommendations

by

Osama MOHAMMED
BSc, Lakehead University

Supervised by
Dr. Rachid Benlamri
Professor and Chair,
Department of Software Engineering

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Master of Science in Electrical and Computer Engineering

in the Faculty of Engineering

© Osama Mohammed, 2012
Lakehead University

All rights reserved. This thesis may not be reproduced in whole or in part, by photocopy or other means, without the permission of the author.

Abstract

There is a growing realization that healthcare is a knowledge-intensive field. The ability to capture and leverage semantics via inference or query processing is crucial for supporting the various required processes in both primary (e.g. disease diagnosis) and long term care (e.g. predictive and preventive diagnosis). Given the wide canvas and the relatively frequent knowledge changes that occur in this area, we need to take advantage of the new trends in Semantic Web technologies. In particular, the power of ontologies allows us to share medical research and provide suitable support to physician's practices. There is also a need to integrate these technologies within the currently used healthcare practices. In particular the use of semantic web technologies is highly demanded within the clinicians' differential diagnosis process and the clinical pathways disease management procedures as well as to aid the predictive/preventative measures used by healthcare professionals.

This thesis is a research attempt to employ various semantic web technologies to support 'as needed' data integration to aid clinicians in the process of disease diagnosis. This thesis describes a novel approach for supporting medical disease diagnosis by making recommendations through semantic based modeling of medical knowledge and inference making. The developed approach defines an ontology model able to represent relations between medical diseases and their signs and symptoms in two related semantic web components: the *evidence-based* and *proximity-based*. These two components utilize two ontologies specially developed for the purpose of disease diagnosis recommendation: the Diseases Symptoms Ontology (DSO) and the Patient Ontology (PO).

The evidence-based component identifies unique semantic web architecture for aiding clinicians to arrive at accurate disease diagnosis using the process of differential diagnosis (DDx). The core structure of the evidence-based component is composed of two main systems. The first system is a DSO crawler which provides clinicians with essential queries to eliminate irrelevant diseases and arrive at a correct diagnosis. The evidence-based model is also composed of a unique rule-based inferential engine employing clinical pathways rules. The DSO uses variety of semantic web technologies including the Jena OWL framework along with a relational query component that replaces the traditional SPARQL query engine for the purpose of DDx recommendation.

The proximity-based component employs data mining techniques for providing predictive diagnosis recommendations besides using similar semantic web technologies used at the evidence-based component (ontology crawler). The proximity-based component analyzes previously available clinical cases to predict the diagnosis of new cases. The proximity-based component provides clinicians with diagnostic recommendations based on classification algorithms as well as identifies new diagnostic rules via the use of association algorithms. Moreover, this thesis describes how these two components can be integrated where the evidence-

based component provides continuous data that will be used by the proximity-based component. Conversely, the proximity-based component identifies further interesting diagnostic rules that can be incorporated into the evidence-based component's rules. Finally, this thesis describes a prototype for testing the design concepts addressed by this thesis.

Dedication

I would like to dedicate this research thesis to my parents, grandparents and family for all the love and support they have given me throughout my research journey and throughout my life.

Acknowledgement

I would acknowledge and thank my supervisor, Dr. Rachid Benlamri, for the valuable help and guidance he has given me throughout my research journey. I would like also to thank the faculty members of the Department of Software Engineering and the Northern Light Research Group for their encouragement and help. Finally I would like to thank Lakehead University for their kind support in providing me with the excellent environment to conduct research.

Table of Contents

Chapter 1: Introduction	8
1.1. Problem Statement and Motivation	8
1.2. Differential Diagnosis and Clinical Pathways	9
1.2.1 Differential Diagnosis	10
1.2.2 Clinical Pathways.....	10
1.2.3 Combining Clinical Pathways and Differential Diagnosis (DDx)	11
1.3. The Role of Semantic Web in Diagnosis Recommendation	11
1.4. Summaries of the Chapters.....	14
Chapter 2: Disease Diagnosis Ontology Engineering	16
2.1. The Disease Diagnostic Dilemma	16
2.2. Ontology Guided Disease Diagnosis	17
2.3. Disease Diagnosis Ontology Engineering	19
2.3.1 The TMO Disease Diagnosis Ontology	21
2.3.2 The Galen Disease Diagnosis Ontology	21
2.3.3 The HPO Disease Diagnosis Ontology	22
2.3.4 The IDO Disease Diagnosis Ontologies.....	22
2.3.5 The DOID Disease Ontology	23
2.3.6 The SYMP Symptoms Ontology.....	24
2.3.7 The GHDO Ontology.....	24
2.4 DSO Ontology: Aligning of DOID and SYMP Ontologies.....	25
Chapter 3: Developing a DSO OWL Ontology Crawler for Disease Diagnosis.....	33
3.1. Ontology Management: Toward Filtering Relational Information	33
3.2. The Basic Infrastructure for Programming the Ontology Crawler	34
3.3. Developing the DSO Crawling Relational Primitives	35
3.3.1 The R1 Relation Description.....	38

3.3.2 The R2 Relation Description.....	38
3.3.3 The R3 Relation Description.....	42
3.3.4 The R4 Relation Description.....	44
3.3.5 The R5 Relation Description.....	45
3.3.6 The R6 Relation Description.....	46
3.4. Enhancing the Power of the Ontology Crawler: The Integration of Pellet Reasoner	47
Chapter 4: Rule-based and Proximity-based Differential Diagnosis Recommenders.....	52
4.1. Disease Diagnosis: Evidence-based vs. Proximity-based	52
4.2. Incorporating Clinical Pathways for Medical Diagnosis Recommendation	57
4.3. Incorporating Proximity Rules for Medical Diagnosis Recommendation	60
4.4. Selecting a Rule Engine for Medical Diagnosis Recommendation.....	65
4.5. The Drools Rule Engine	66
4.5.1 Using the Drools Rule Engine	66
4.5.2 Drools Rule Firing Mechanism	66
4.6. Medical Data & Data Pre-processing for Proximity-Based Diagnosis Recommendation	71
4.6.1 Selecting Suitable Medical Dataset for Proximity-Based Diagnosis Recommendation	71
4.6.2 Data Pre-processing, Transformation and Filtering.....	73
4.7. Weka Data Mining Tool for Proximity-Based Diagnosis Recommendation.....	76
4.8. Testing the Proximity-Based Diagnosis Recommender	76
4.9. Validation of Evidence & Proximity-based DDx Recommendations.....	77
4.9.1 Evidence-based vs. Proximity-based Rules Comparison.....	77
Listing 4.3: Proximity-based Rule for Diabetes	79
4.9.2 Evidence-based vs Proximity-based Predictions.....	80
Chapter 5: System Demonstration.....	82
5.1. Clinical Diagnosis Support Systems.....	82

5.2. Prototyping our Differential Diagnosis Recommender.....	84
5.3. Validation Scenarios for the DDx Recommendation Prototype	85
5.3.1: Validation of the DDx Recommender’s DSO Ontology Crawler Relations.....	85
5.3.2: Validation Scenario for the Evidence-based DDx Recommender.....	90
5.3.3. Validation of the Proximity-based DDx Recommender	99
Chapter 6: Conclusions and Future Research	108
6.1. Conclusions	108
6.2. Future Research	112
Research Publications	115
References	116
Appendix A.....	126
Using Weka for Classification.....	139
4.9.2 Association Rules	148
Appendix B	150
Appendix C	165
Appendix D.....	170
Appendix E	175

Chapter 1: Introduction

1.1. Problem Statement and Motivation

Recent advances in medical technology have significantly improved human health in many countries like Canada. However, these advances remain out of touch for much of the world population. We face unprecedented healthcare challenges in the 21st century. The health care industry is facing key changes and challenges which could dramatically affect healthcare provision (see Table 1.1). Thus, software engineers are expected to play a critical role in developing novel and affordable medical technology to solve global healthcare problems [Akay 2008].

Central to all these challenges is the issue of sensitive diagnosis and rapid treatment. In our current world, there are many different diseases where some are harder for doctors than others. There are some diseases that are often difficult to correctly diagnose. Any type of disease that is rare is often hard to diagnose, simply because doctors may not be familiar with its details¹. There are many other examples of diseases that are hard to diagnose. The digestive system as an example has somewhat vague symptoms and ones that patients are often embarrassed about. It is also difficult to look inside the different parts of the digestive tract. Hence, it is difficult to diagnose between the various digestive disorders such as: Bacterial or viral infection of the digestive (e.g. infectious diarrhoea), Food poisoning, Irritable Bowel Syndrome (IBS), Crohn's disease, Ulcerative colitis, Celiac disease (gluten sensitivity), Diabetic gastroparesis (stomach nerve neuropathy), and Diabetic diarrhoea (intestine nerve neuropathy).

Moreover, there are over-diagnosed conditions and under-diagnosed conditions. There are certain diseases that get over-diagnosed more often than others. This means that the doctor gives this disease as the diagnosis, when in fact there is some other cause or disease. Some common examples include²: **Irritable Bowel Syndrome (IBS), Middle ear infection (acute otitis media) in children, Lyme disease, Alzheimer's disease.**

Under-diagnosis is common for conditions that have either no symptoms or only vague or mild symptoms. Under-diagnosis can also occur for conditions that are rarer than other conditions and thus simply don't get considered by patients and their doctors. Examples on under-diagnosed diseases include³: **High cholesterol, Hypertension, Infectious diarrhea, Lactose intolerance, Celiac disease.**

¹ <http://www.rightdiagnosis.com/intro/difficult.htm>

² www.rightdiagnosis.com/intro/overdiag.htm

³ <http://www.medlabstats.com/students/Over-under-diagnosis.pdf>

Table 1.1: Global Healthcare Challenges³

Global economics	Increasing affluence in emerging economies sees increasing demand for integrated healthcare provision
Demographics	The proportion of people aged over 65 will increase from 7.3% to 9.4% of the overall population by 2020. The demands of age-specific ailments will place increasingly difficult demands on healthcare services.
Epidemiological trends	Genetics, diet and environmental factors all impinge on the prevalence and severity of disease affecting different populations. Increasing affluence in developing countries will see increases in chronic diseases associated with environment, age and lifestyle. For example, the World Health Organisation estimates that the number of obese people worldwide will increase from 400 million in 2005 to over 700 million by 2015.
Pharmacogenomics	Genetic variations also affect the way individuals respond to drug treatments. The responses can have varying consequences, ranging from poor responses to therapy to quite severe side effects. There is an increased need to understand the basis of these variations in order to improve the development of new treatments as well as patient care.
Environmental changes	The impact of climate change on human health is difficult to predict; however, an increase in prevalence or geographical reach of vector-borne diseases (eg malaria and sleeping sickness) is highly likely. Large health effects are also likely from food supply changes, environmental degradation and population movements.

Thus the process of diagnosing diseases is challenging and imprecise, as the symptoms and signs vary widely. For this reason, differential diagnosis (DDx) (see section 1.2.1 below) is a common practice in medicine.

This thesis attempts to develop various semantic web methods and techniques for providing recommendations for clinicians to assist them in the process of DDx. The methods and techniques developed in this thesis integrate both practical medical knowledge using clinical pathway rules for a set of common diseases as well as effective semantic web technologies (such as the development of a disease symptom ontology, development of patient ontology, using rule-based recommenders based on clinical pathways and data mining).

1.2. Differential Diagnosis and Clinical Pathways

At many times, an exact diagnosis can be determined using a hybrid approach involving clinical pathways along with DDx [Colucciello *et al.* 1999]. In this hybrid approach, clinicians use the clinical pathways as a guide to narrow a list of possible diagnosis paths predicted using DDx, for a specific patient case, down to the correct diagnosis path(s).

³ Medicines and Healthcare Strategy 2009-2012: http://www.innovateuk.org/_assets/pdf/Corporate-Publications/MedicineHealthcareExecSum.pdf

1.2.1 Differential Diagnosis

DDx is a systematic method used to identify unknowns. This method, essentially a process of elimination, is used by physicians, nurse practitioners, physician assistants, and other trained medical professionals to diagnose a specific disease in a patient [Wisconsin Fibromyalgia Network 2011]. It often involves first making a list of possible diagnoses, then attempting to remove diagnoses from the list until one diagnosis remains [Cray 2011]. In some cases, there will remain *no* diagnosis; this suggests the physician has made an error, or that the true diagnosis is unknown to medicine. Removing diagnoses from the list is done by making observations and using tests that should have different results, depending on which diagnosis is correct.

Differential diagnosis allows the physician to:

- more clearly understand the condition or circumstance the patient is suffering from
- assess reasonable prognosis
- eliminate any imminently life-threatening conditions
- plan treatment or intervention for the condition or circumstance
- enable the patient and the family to integrate the condition or circumstance into their lives, until the condition or circumstance may be ameliorated, if possible

DDx is often manual and requires the estimation of multiple distinct parameters in order to determine the most probable diagnosis. For this reason, there are many attempts to use computer-based DDx software (e.g. ODDIN [García-Crespo *et al.* 2010], MEDBOLI [Rodriguez *et al.* 2009]) to automate the process of diagnosis and increase its accuracy. Building a computer-based differential diagnosis system implies using a number of knowledge-based technologies which avoid ambiguity, such as ontologies representing specific structured information, but also strategies such as computation of probabilities of various factors and logical inference, whose combination outperforms similar approaches [García-Crespo *et al.* 2010].

1.2.2 Clinical Pathways

Every healthcare institution uses documented "best practices" interventions and therapy standards when managing specific disease processes. Such best practices represent a set of "optimal" management models for a certain diagnosis and treatment (therapy), which are generally termed as "Clinical Pathways". Clinical pathways, also known as care maps, are designed to be used in conjunction with the present standard of care as a tool to decrease variation in outcomes and maintain care within a specified community of practice. Clinical pathway programs aims to optimize clinical and economic outcomes for disease management by doing the following [Rossi 2003]:

- facilitating proper diagnosis
- maximizing clinical effectiveness
- eliminating ineffective diagnosis and therapeutic procedures
- maximizing the efficiency of care delivery

1.2.3 Combining Clinical Pathways and Differential Diagnosis (DDx)

Clinical pathways assist to find solutions when differential diagnosis may face difficulties in sorting an overwhelming and confusing set of signs and symptoms. Imagine a physician in an emergency department trying to diagnose a critically ill infant who has feeding problems (vomiting, spitting up, refusing to feed), poor urine output, altered mental status (fussiness, somnolence, or unresponsiveness), respiratory difficulties (increased work of breathing, respiratory fatigue/failure), and temperature abnormalities (high or low) [Brown et al 2002]. Table 1.2 illustrates the differential diagnosis options for critically ill infants. Using differential diagnosis alone may provide very long list of possible diagnoses in some cases. The integration between DDx and clinical pathways requires a higher level of IT technology that utilizes the notion of semantic web [Niekerk & Griffiths 2008, Luc et al 2003] as well as support for incorporating emerging technologies such as data mining [Soni et al 2011].

1.3. The Role of Semantic Web in Diagnosis Recommendation

Generally, disease diagnosis processes heavily depend on both information and knowledge. Information systems are typically integrated into hospitals to support organization processes such disease management and result reporting, etc. Although medical databases and information management systems are common, healthcare knowledge, which is important for medical diagnosis and treatment, is rarely integrated into software systems supporting healthcare processes [Buranarach *et al.* 2009]. The semantic web is a concept that involves incorporating descriptions into data to make the data reusable and enable applications to be built that can take advantage of this describable collection of data. It features a common framework for data to be shared and reused across application, enterprise, and community boundaries. This opens a new set of opportunities that can be utilized to improve health care management. However, the support for semantic interoperability across a large number of sub-domains requires that rich, machine-understandable descriptions are consistently represented by well formulated vocabularies drawn from formal ontologies and that they can be easily composed and published by domain experts [Dumontier 2010]. For this purpose, our thesis starts by developing an ontology-based knowledge management framework that focuses on providing information and knowledge support for knowledge-enabled diagnosis services.

Table 1.2: DDx Possible Diagnosis for Critically Ill Infants [Brown *et al.* 2002]

<p>Infectious and respiratory Most serious:</p> <ul style="list-style-type: none"> • Sepsis • Meningitis/Encephalitis <p>Most common:</p> <ul style="list-style-type: none"> • Bronchiolitis/Asthma <p>Others:</p> <ul style="list-style-type: none"> • Infant botulism • Bacterial tracheitis/severe croup • Pertussis <p>Cardiac</p> <ul style="list-style-type: none"> • Congenital heart disease • Myocarditis • Supraventricular tachycardia <p>Genitourinary</p> <ul style="list-style-type: none"> • Congenital obstructive lesions <p>Gastrointestinal Most serious:</p> <ul style="list-style-type: none"> • Malrotation with midgut volvulus 	<p>Most common:</p> <ul style="list-style-type: none"> • Gastroenteritis/Dehydration • Intussusception • Incarcerated inguinal hernia <p>Others:</p> <ul style="list-style-type: none"> • Hirschsprung's disease • Meconium plug • Bowel atresia • Appendicitis <p>Toxicologic</p> <ul style="list-style-type: none"> • Carbon monoxide poisoning • Methemoglobinemia • Drug toxicity (see Table 3) <p>Endocrinologic/Metabolic Congenital adrenal hyperplasia</p> <ul style="list-style-type: none"> • Acute salt wasting crisis <p>Hyponatremia</p> <ul style="list-style-type: none"> • Inappropriate formula preparation <p>Diabetic ketoacidosis</p>	<p>Hypoglycemia</p> <ul style="list-style-type: none"> • Secondary to inborn errors of metabolism • Lack of intake • Insulin/Food mismatch <p>Neurologic Most serious:</p> <ul style="list-style-type: none"> • Ventriculo-peritoneal shunt failure/infection • Acute hydrocephalus <p>Most common:</p> <ul style="list-style-type: none"> • Seizures <p>Others:</p> <ul style="list-style-type: none"> • Arteriovenous malformation • Central nervous system tumor • Nonconvulsive status epilepticus <p>Traumatic Non-accidental trauma ("shaken-baby syndrome")</p>
---	--	---

Our approach starts by developing a diseases symptoms ontology (DSO) for diagnosing chronic diseases that are common in any primary care healthcare unit. We have selected about a dozen diseases to include in our DSO ontology. Currently, there are no existing DSO ontologies in existence. However, there are separate ontologies for diseases and symptoms. We propose to build on these existing ontologies to create a model for the DSO ontology. A more detailed discussion of ontology as a knowledge structure, and a more detailed discussion of our DSO model is the subject of chapter 2. Once a DSO ontology knowledge structure is developed, outside entities/systems need to be able to make use of this knowledge structure. In other words, a query engine that has the ability to retrieve specific information (queries) from the DSO knowledge structure must be developed. The query engine will allow other systems to retrieve information from the DSO. We propose to use the Jena Java API to construct our query engine. This will be discussed in detail in chapter 3. We will also discuss in that chapter the design and implementation of common queries in differential diagnosis (DDx) as far as the relations between symptoms and diseases are concerned. We will call our query engine the DSO crawler. Another important knowledge support for disease management is various types of patient related information. Such patient attributes are mentioned in various medical documents. Examples of patient related attributes include age, weight, height, etc. This information is specifically important for disease diagnosis. Therefore, we also propose to include patient related information into an ontology knowledge structure. We call this ontology a patient ontology (PO). We will discuss the PO ontology and its query engine (PO crawler) in detail in chapter 4. On top of the DSO & PO crawlers, we will employ a flexible rule-based engine that can accommodate various clinical pathway rules used by the disease management process to recommend tests, procedures, and/or to provide a diagnosis. The clinical pathway rules should model the rules of differential diagnosis for the common diseases we have selected. A more detailed discussion of clinical

pathway and ontology driven differential diagnosis recommendation will be included in chapter 4.

Another approach for differential diagnosis recommendation is a data mining approach. Like the above approach, it is an ontology driven approach based on the DSO & PO crawlers. Unlike the above approach, it is based on data mining techniques rather than clinical pathways rules. The DSO & PO crawlers select the most important medical attributes, for a specific diagnostic case, from a patient dataset and feed these attributes to data mining algorithms. The data mining algorithms then process the data and come to certain diagnostic conclusions. This data mining and ontology driven differential diagnosis recommendation approach is discussed further in chapter 4. Moreover, the two approaches can cooperate to form an overall differential diagnosis recommendation approach. We call the overall approach the overall differential diagnosis recommendation approach and we will discuss it further in chapter 4. Figure 1.1 illustrates our overall differential diagnosis recommendation model.

We followed that by querying the DSO ontology according to some common questions used for differential diagnosis to eliminate or confirm some diagnosis options. For this purpose we designed our own DSO crawler utilizing some generic relations that provide answers to the required queries. On top of the DSO crawler we developed our flexible rule-based engine that can accommodate the various clinical pathway rules used by the disease management process to recommend tests, procedures, and/or to provide a diagnosis. Our flexible rule-based recommender is able to store every approved recommendation as a training dataset. The training dataset can be used by clinicians to predict the diagnosis of future cases using some sound data mining techniques and algorithms. Our developed DDx recommender uses our developed patient ontology to guide mining new diagnostic trends and pattern in patient data. Figure 1.1 illustrates our overall development approach.

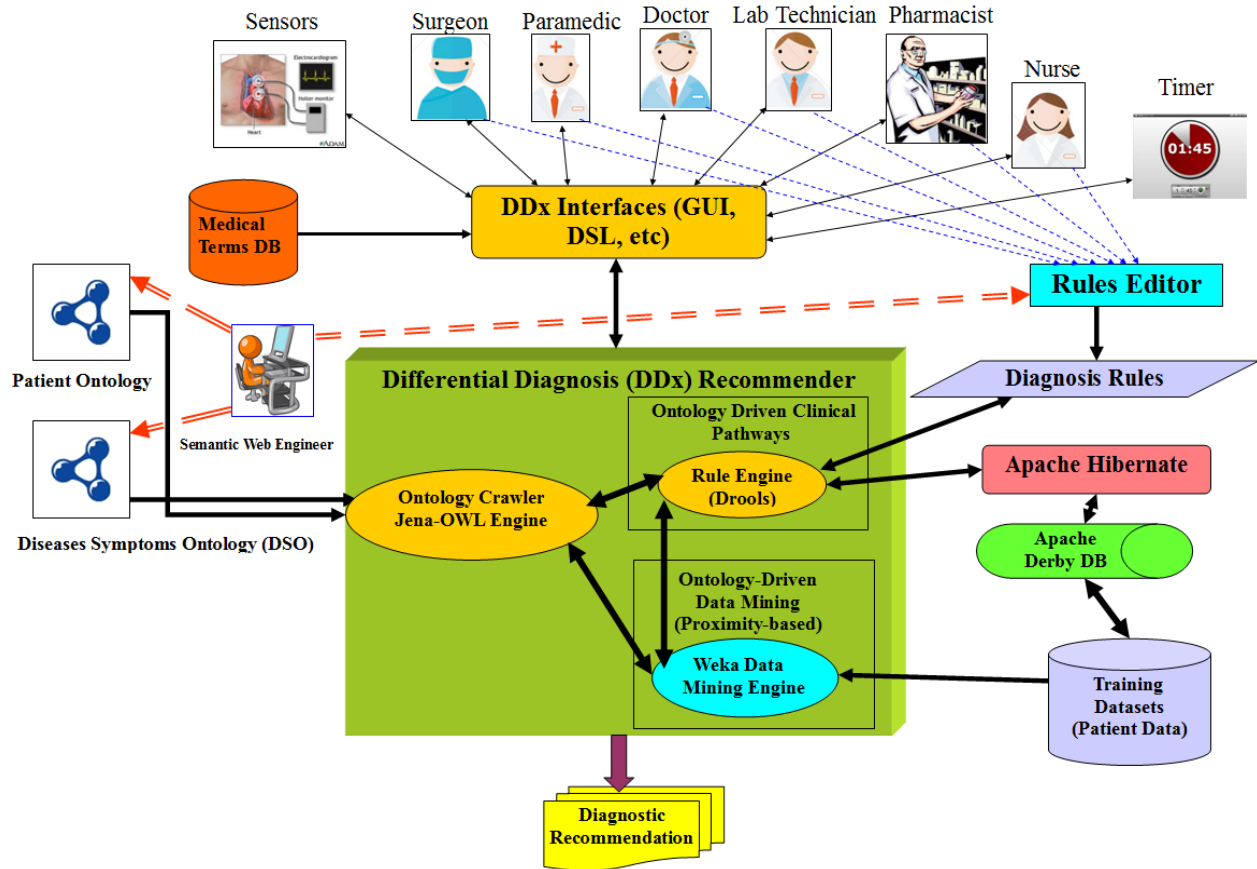


Fig. 1.1: Overall Architecture for the DDx Recommender

1.4. Summaries of the Chapters

Chapter 1 introduced essential concepts in our thesis such as DDx, Clinical Pathways, and Semantic Web technologies for DDx Recommendation. It also summarized our thesis problem statement. **Chapter 2** will develop an ontology that relates symptoms to diseases. We call this ontology the diseases symptoms ontology (DSO). DSO is a hybrid ontology combining two OBO standard ontologies (DOID & SYMP). **Chapter 3** will present a unique architecture for querying the DSO based on the notion of DDx. We call this architecture the DSO Ontology Crawler. Based on this crawler, clinicians are able through our six relational queries to answer representative DDx questions towards narrowing diagnosis options and providing primitive recommendation. **Chapter 4** will develop another ontology called the patient ontology (PO) along with its crawler. It also develops two recommendation components based on interacting with clinicians and semantic web technologies. The first component introduces the notion of flexible rule-based engine that can accommodate various clinical pathway rules. It is a clinical pathway and ontology driven differential diagnosis recommendation model. The second component utilizes previous patient's data to predict diagnosis for new patient cases. It is a data mining and ontology driven differential diagnosis recommendation model. Chapter 4 ends by

introducing an overall semantic web based differential diagnosis recommendation framework that integrates the two recommendation components. **Chapter 5** will introduce our experimentation and scenarios of using our integral recommendation framework prototype. Finally, **chapter 6** summarizes our thesis conclusions and the future research that can build on our research. In addition, it discusses challenges of implementation of our integral recommendation framework. It then suggests some practical diagnosis recommendation applications that can be developed using our integral recommendation framework.

Chapter 2: Disease Diagnosis Ontology Engineering

2.1. The Disease Diagnostic Dilemma

Many medical diagnosis systems are proposed in the medical domain to help solve the disease diagnosis problem. Table 2.1 lists some of the existing autonomous disease diagnoses systems. However, such systems reported limited acceptance and success in the clinical practice because of their lack of scalability as well as the complex and the dynamic nature of the medical diagnosis process. Diagnosis is a very complex process as it involves identifying symptoms, signs and results of different diagnostic procedures (such as medical tests) as well as involving a higher level of decision making based on complex knowledge repositories and expert interactions between health professionals. In reality, a physician formulates a hypothesis of probable diagnoses, and, in many cases, will obtain further testing to confirm or clarify the diagnosis before providing treatment. The diagnosis process may thus continue for a number of iterations before the patient is finally diagnosed with enough certainty and the cause of the symptoms is recognized. For this reason a computer program used in the diagnostic procedure must incorporate all the dynamic issues and the required knowledge and semantics of the disease diagnosis process. Such systems are support systems as they are clearly designed to interact with health professionals and support decision making by health professionals. Figure 2.1 illustrates a simple mindmap used for diagnosing Type 2 Diabetes [Schwimmer 2007].

Table 2.1: Some Medical Diagnosis Support Systems

Diagnosis System	Purpose	Reference
CDSS	To Diagnose babies in neonatal intensive care unit.	[Catley <i>et al.</i> 2003]
DXplain	Clinical decision support expert system that explains how it obtains its diagnosis recommendations and therefore can be used as an educational medical reference system.	[Barnett <i>et al.</i> 1987]
MADHS	A multi agent diagnosis helping system.	[Yang and Shieh 2008]
RBDDS	A temporal system to express temporal relationships among diseases that may have mutual affect potentially	[Chien-Chih <i>et al.</i> 2007]
PROMEDAS	Patient Specific Clinical Diagnostic Support System	[Kappen <i>et al.</i> 2003]

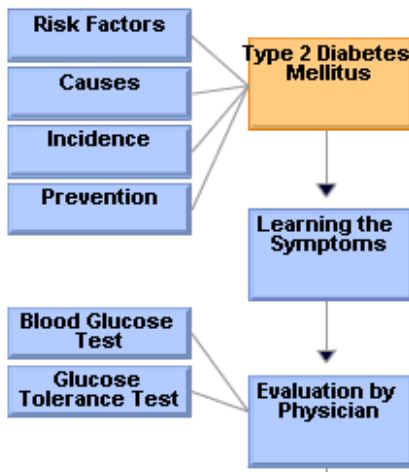


Fig. 2.1: Mindmap for Type 2 Diabetes Diagnosis [Schwimmer 2007]

2.2. Ontology Guided Disease Diagnosis

Development of knowledge structures for healthcare is a research community-wide effort focused on the development of a set of interoperable knowledge modules that together provide solutions to many healthcare challenges of the 21st century. Such efforts encompass a wide range of activities, including ontology engineering, biomedical modeling, data mining, knowledge discovery tools and database development, simulation, and visualization. However, effective knowledge representation requires the use of standardized vocabularies to ensure both shared understanding between people and interoperability between information systems. Internationally, there are countless existing biomedical vocabularies such as SNOMED-CT⁴, LOINC⁵, ICD-9 CM⁶, MeSH⁷ and UMLS⁸. Unfortunately, many of these existing biomedical vocabulary standards rest on incomplete, inconsistent, or confused accounts of basic terms pertaining to diseases, diagnoses, and clinical phenotypes [Scheuermann 2009]. There is no one universally accepted coding scheme that encapsulates all pertinent clinical information for the purposes of patient care, clinical research and disease diagnosis. There are indeed several attempts to harmonize such terminologies but such efforts are at their infancy [Hamm *et al.* 2007]. Without a widely accepted clinical and medical terminology (nomenclatures, thesauri, classifications, etc.), disease identification and reporting remains mission incomplete especially when we consider that interpersonal communication is an essential activity in disease diagnosis. Solving this problem requires the effort of notable institutions to develop concepts and

⁴ <http://www.openclinical.org/medTermSnomedCT.html>

⁵ <http://loinc.org/>

⁶ <http://icd9cm.chrisendres.com/>

⁷ <http://www.ncbi.nlm.nih.gov/mesh>

⁸ <http://www.nlm.nih.gov/research/umls/>

classifications dedicated for disease identification in order to share information in a semantically unambiguous way, and to reuse domain knowledge. Moreover we cannot rely on the human brain of the clinical experts to remember all diseases' details since the number of diseases which exist worldwide is enormous. For this reason we need an effective way to store and retrieve knowledge related to human diseases. In this direction ontologies play a crucial role in defining concepts and in establishing relations between these concepts. Ontologies are an important phase in the process of knowledge base development and management of any modern clinical system. They are dynamic and keep maturing over time due to changes in the local environments. Medical ontologies are much more than biomedical vocabularies. They arrange concepts into ISA and sibling hierarchies, which effectively relate these concepts in a structural way that provides valuable inferences upon retrieval.

Two general strategies for ontology development are predominant, one is based on using sophisticated ontology modelling tools such as Protégé [Stanford 2011], Ontolingua⁹, Chimaera¹⁰ and LOOM¹¹ and the second one is based on inferential programming and logical reasoning. Although both approaches may be used at the same time, the advent of expressive ontology languages such as OWL¹² and its close relation to Description Logics (DL)¹³, non-trivial implicit information, such as the **is-a** hierarchy of classes, can often be made explicit by OWL-API¹⁴ inferential programming methods or through the use of logical reasoners¹⁵. Figure 2.2 illustrates the way ontology may define the concept of Diabetes Type 1 disease.

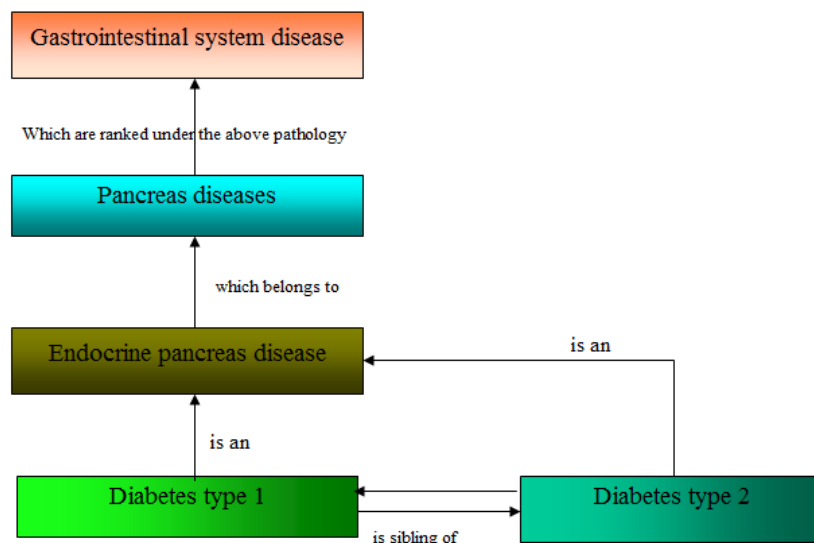


Fig. 2.2: Disease Ontology ISA and Sibling Hierarchies Relate and Classify Diseases

⁹ <http://www.ksl.stanford.edu/software/ontolingua/>
¹⁰ <http://www.ksl.stanford.edu/software/chimaera/>
¹¹ <http://www.bioontology.org/wiki/index.php/LOOM>
¹² <http://www.w3.org/TR/owl-ref/>
¹³ http://en.wikipedia.org/wiki/Description_logic
¹⁴ <http://owlapi.sourceforge.net/>
¹⁵ http://en.wikipedia.org/wiki/Semantic_reasoner

Common components of ontologies include¹⁶:

- **Individuals:** instances or objects (the basic or "ground level" objects)
- **Classes:** sets, collections, concepts, classes in programming, types of objects, or kinds of things
- **Attributes:** aspects, properties, features, characteristics, or parameters that objects (and classes) can have
- **Relations:** ways in which classes and individuals can be related to one another
- **Function terms:** complex structures formed from certain relations that can be used in place of an individual term in a statement
- **Restrictions:** formally stated descriptions of what must be true in order for some assertion to be accepted as input
- **Rules:** statements in the form of an if-then (antecedent-consequent) sentence that describe the logical inferences that can be drawn from an assertion in a particular form
- **Axioms:** assertions (including rules) in a logical form that together comprise the overall theory that the ontology describes in its domain of application. This definition differs from that of "axioms" in generative grammar and formal logic. In those disciplines, axioms include only statements asserted as a priori knowledge. As used here, "axioms" also include the theory derived from axiomatic statements
- **Events:** the changing of attributes or relations

2.3. Disease Diagnosis Ontology Engineering

Ontology engineering (or ontology building) is a field that studies the methods and methodologies for building ontologies. It studies the ontology development process, the ontology life cycle, the methods and methodologies for building ontologies, and the tool suites and languages that support them. It aims to make explicit the knowledge contained within software applications, and within enterprises and business procedures for a particular domain. Ontology engineering offers a direction towards solving the interoperability problems brought about by semantic obstacles, such as the obstacles related to the definitions of business terms and software classes. Ontology engineering is a set of tasks related to the development of ontologies for a particular domain [Maniraj and Sivakumar 2010][De Nicola 2009]. For the last few decades software engineers and scientists have been exploring ways of 'modeling' or 'representing' the entities about which computers are expected to reason. But what do 'modeling' and 'representing' mean? What is a 'conceptual model' or an 'information model' and how can they and their components be unambiguously described? The problem of multiple conflicting meanings arises also in regard to other terms, such as 'class', 'object', 'instance', 'individual', 'property', 'relation', etc., all of which have established, but unfortunately non-uniform, meanings in a range of different disciplines including clinical diagnosis. In OWL, 'instance' means 'element' or 'member' of a class. A number of influences have played a role in the

¹⁶ http://en.wikipedia.org/wiki/Ontology_components

development of terminology resources for applications in medicine [Smith and Brochhausen 2011]:

- The influence of library science and of dictionary and thesaurus makers, illustrated mostly by MeSH, the indexing resource maintained by the National Library of Medicine;
- The influence of database design and conceptual modeling, illustrated for example by the HL7 initiative¹⁷;
- The influence of biological science, illustrated by the Gene Ontology (GO)¹⁸ and by the other ontologies within the Open Biomedical Ontologies (OBO) Foundry initiative¹⁹ and the National Center for Biomedical Ontology²⁰;
- The influence of advances towards greater formal rigor, illustrated for example by current developments within SNOMED-CT and within the framework of the Semantic Web

Several efforts by a variety of standardization agencies such as ISO, CEN and W3C have been undertaken to provide cross-disciplinary uniformity but with little or no success reported. The only notable result that can be positively identified is the one by the OWL-DL²¹ semantic web community where it has a rigorously defined semantics. This does not mean, however, that OWL-DL guarantees that an ontology formulated using OWL-DL is an error-free representation of its intended domain [Smith and Ceusters 2006]. Related to the goal of outlining a terminological framework that encompasses diseases, their causes and manifestations, and diagnostic acts, Table 2.2 lists some notable efforts in this direction. A careful inspection to the attempts listed in Table 2.2 reveals that such entities have not been adequately treated in standard vocabulary resources [Scheuermann 2009].

Table 2.2: Notable Dedicated Disease Diagnosis Ontologies

<i>Disease Diagnosis Ontology</i>	<i>Purpose</i>	<i>URL</i>
(TMO) Translational Medicine Ontology	Integrate data across aspects of drug discovery and clinical practice.	http://code.google.com/p/translationalmedicineontology/
OpenGalen/Galen Ontologies	Terminology servers and data entry systems based on a Common Reference, or CORE, model for medical terminology.	http://www.opengalen.org www.co-ode.org/galen/
Phenomizer (HPO)	Human Phenotype Ontology	http://www.human-phenotype-ontology.org/index.php/hpo_browserse.html
Infectious Disease Ontologies (IDO)	The IDO ontologies are designed as a set of interoperable ontologies that will together provide coverage of the infectious disease domain.	http://infectiousdiseaseontology.org/page/Main_Page

¹⁷ www.hl7.org

¹⁸ <http://www.geneontology.org/>

¹⁹ <http://www.obofoundry.org/>

²⁰ <http://www.bioontology.org/>

²¹ http://semanticweb.org/wiki/OWL_DL

2.3.1 The TMO Disease Diagnosis Ontology

TMO is a high-level, patient-centric ontology that extends existing domain ontologies to integrate data across aspects of drug discovery and clinical practice. The ontology has been developed by participants in the W3C Semantic Web for Health Care and Life Sciences Interest Group and members of the National Center for Biomedical Ontology²². The ontology is available in OWL format²³. The ontology enables silos in discovery research, hypothesis management, experimental studies, compounds, formulation, drug development, market size, competitive data, population data, etc. to be brought together. This will help pharmaceutical companies to model patient-centric information, which is essential for the tailoring of drugs, and for early detection of compounds that may have suboptimal safety profiles²⁴.

2.3.2 The Galen Disease Diagnosis Ontology

GALEN is a project for developing terminology servers and data entry systems based on a Common Reference, or CORE, model for medical terminology [Rector 1996]. It is an attempt to represent clinical concepts using a logic-based formalism based on certain choices. These choices are embodied in the high level schemas of the resulting logical model or ontology. The GALEN Common Reference Model source files are available in both original GRAIL²⁵ notation and also in OWL-RDF from the OpenGalen portal²⁶. A key feature of the GALEN approach is that it divides the problem of clinical knowledge representation and terminology into distinct parts each implemented by different software:

- a) The concept representation, or ontology, schema and model expressed in GRAIL.
- b) The linguistic resources for presenting the model
- c) The mappings to and from the concept model and other representations
- d) Perspectives, views and intermediate representations of the model which adapt it to particular purposes
- e) Indexes to other knowledge based on the model
- f) Non-terminological computational or other reasoning mechanisms, e.g. unit conversion
- g) The terminology server and its API which make all of the other parts available as a coherent whole to applications and users.

GALEN's original idealized goal was an ontology which could express 'all and only' what was medically sensible. It was recognised from the start that this ideal was unobtainable and that the 'all' would have to take precedence over the 'only' because [Rector and Rogers 1999]:

²² <http://code.google.com/p/translationalmedicineontology/>

²³ <http://translationalmedicineontology.googlecode.com/svn/trunk/ontology/tmo.owl>

²⁴ <http://bioportal.bioontology.org/ontologies/1461>

²⁵ <http://www.opengalen.org/sources/sources.html>

²⁶ <http://www.opengalen.org/sources/sources.html>

- a) There are well known trade-offs between expressiveness and computational tractability in formal systems.
- b) Reality is fractal – no matter how much detail is represented in the model, it is always possible to represent more
- c) Constraints on what is ‘sensible’ are slippery and difficult to formulate loosely enough to allow all sensible statements but strictly enough to exclude all nonsense
- d) The practical goal has been to have an ontology which is sufficiently expressive to capture the statements and abstractions used by clinicians and classify them correctly while rejecting patent nonsense.

To use Galen for disease diagnosis there is need for complicated translation system like MoST (Model Standardisation using Terminology) Systems. The MoST system developed for this purpose aims to find semantically equivalent SNOMED-CT terminology codes to map to archetype data model fragments. The two key stages of MoST include, (i) term finding, and (ii) data mapping [Qamar 2008]. For this purpose it is very difficult to use the Galen ontology for disease diagnosis due to its complex structure, which is compromised of several module lists above, and which requires complex processing and translation.

2.3.3 The HPO Disease Diagnosis Ontology

Phenomizer is a web-based application²⁷ for clinical diagnostics in human genetics using semantic similarity searches in ontologies [Köhler *et al.* 2009]. One can use the Phenomizer to infer semantic similarity metrics to measure phenotypic similarity between queries and hereditary diseases annotated with the use of the Human Phenotype Ontology (HPO). This can be done by developing a statistical model to assign probability values to the resulting similarity scores, which can be used to rank the candidate diseases. The annotated HPO provides an accurate description of phenotypic abnormalities and therefore provides the foundation of clinical diagnostics and the basis of our understanding of diseases. Unfortunately, HPO and the Phenomizer can only be used to represent hereditary diseases in humans, each of which displays a more or less specific combination of phenotypic features. Therefore it cannot be used for general disease diagnosis purposes.

2.3.4 The IDO Disease Diagnosis Ontologies

The IDO ontologies²⁸ are designed as a set of interoperable ontologies that will together provide coverage of the infectious disease domain. At the core of the set is a general Infectious Disease Ontology (IDO-Core) of entities relevant to both biomedical and clinical aspects of most

²⁷ <http://compbio.charite.de/phenomizer>

²⁸ http://infectiousdiseaseontology.org/page/Main_Page

infectious diseases. Sub-domain specific extensions of IDO-Core complete the set providing ontology coverage of entities relevant to specific pathogens or diseases.

Similar to the Phenomizer, IDO is not general enough to be used for general disease diagnosis.

2.3.5 The DOID Disease Ontology

A human disease ontology is needed to identify all human diseases, provide a hierarchical structure where these diseases are related by parent-child and sibling relationships, and provide information about diseases such as their defining effects on human health. Organising diseases in an ontology hierarchy is extremely useful as it forms a pathological classification of diseases for use in medical systems. Such an undertaking is massive given the number of known human diseases and the fact that new diseases are discovered. The most prominent disease ontology developed to date is the Human Disease Ontology (DOID)²⁹. Started in 2003 as part of the NUGene project at Northwestern University, it has been published in several versions over several years and contains to this date over 8600 known human diseases and 14,600 terms [Northwestern University 2003]. DOID is currently a standard ontology adopted by the OBO Foundry³⁰. There are attempts under way to modify the human disease ontology to include symptoms and relations between those symptoms to diseases in the disease ontology.³¹ In the latest version of the DOID ontology this is clear as a `has_symptom` object property is already defined but no clear use of it has been proposed yet. Figure 2.3 shows the `has_symptom` property under the DOID.

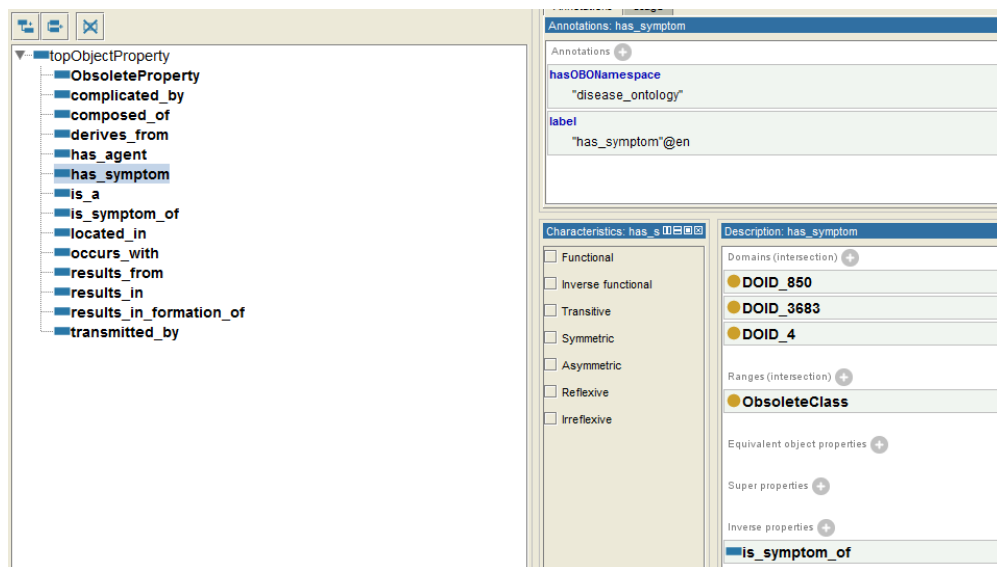


Fig. 2.3: The `has_symptom` Object Property of DOID Ontology

²⁹ http://do-wiki.nubic.northwestern.edu/index.php/Main_Page

³⁰ <http://obofoundry.org/>

³¹ http://do-wiki.nubic.northwestern.edu/index.php/Main_Page

2.3.6 The SYMP Symptoms Ontology

Symptom Ontology (SYMP) [University of Maryland 2005] was developed in 2005 by the Institute for Genome Sciences (IGS) at the University of Maryland, today it contains more than 900 symptoms. SYMP's hierarchy categorizes symptoms under certain headings for example categorizing all types of pain (arm, leg, headache, back pain, chest pain, etc) under physical pain. SYMP became a standard ontology and was adopted by the OBO Foundry during 2008. Figure 2.4 illustrates such categorizing.

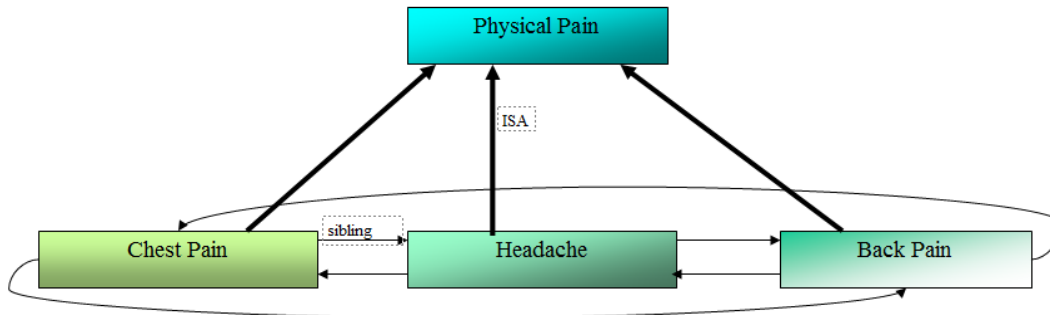


Fig. 2.4: Symptoms Ontology Defines and Relates Symptoms

2.3.7 The GHDO Ontology

The process of diagnosis needs both knowledge of disease hierarchies and symptom hierarchies. Moreover, it also needs relations between diseases and symptoms. For example, what are the symptoms of a certain disease, what diseases have a certain common symptom, what diseases a certain set of symptoms may point to, etc. These relations between symptoms and diseases can be ontologically established. One such proposed ontology model that has not yet been implemented is the GHDO (Generic Human Disease Ontology) [Hadzic and Chang, 2005]. It proposes an ontology model that relates diseases to symptoms (phenotypes) and to the other three elements that uniquely identify a disease: disease type, causes, and treatment. Figure 2.5 shows the GHDO proposed model.

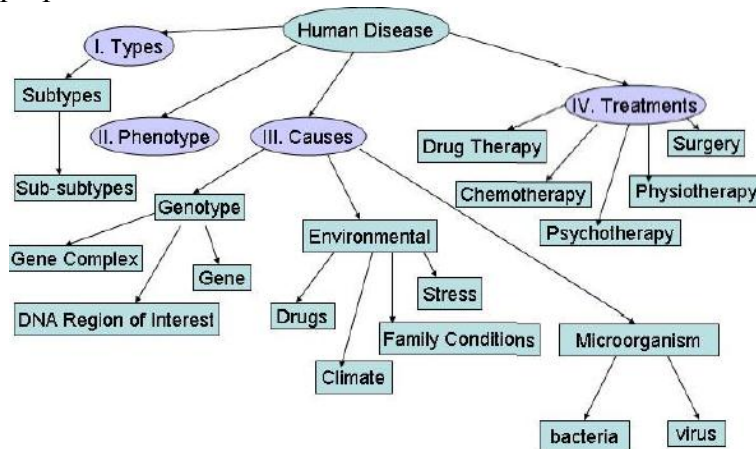


Fig. 2.5: GHDO Model [Hadzic and Chang, 2005]

2.4 DSO Ontology: Aligning of DOID and SYMP Ontologies

Ontologies in the age of semantic web tend to be put everywhere. They are viewed as the silver bullet for many applications. However, in open or evolving systems different parties would, in general, adopt different ontologies and thus growing research on linking or aligning ontologies is considered important [Marques 2005]. Ontology alignment is the idea of combining two (or more) ontologies into one and defining relationships between the concepts of the ontologies forming a new ontology in the process. Alignment between ontologies is a critical challenge for semantic interoperability [Hughes 2004] as well as for producing hybrid ontologies. When a domain is represented by multiple ontologies, there is a need for creating mappings among these ontologies elements in order to facilitate the integration of data and reasoning across these ontologies [Zhang and Bodenreider 2007]. There are two main approaches to alignment: *Ontology Matching* and *Ontology Linking*. Ontology matching techniques are for relating ontologies on the same domain or on partially overlapping domains. For example, ontology mapping works if two disease ontologies are to be aligned. In such case, disease classes from both ontologies are matched. Special mapping constructs are used to indicate how elements from different ontologies are semantically related or equivalent [Doan 2003]. Ontology linking, in contrast, allows elements from distinct ontologies to be coupled with links [Homola and Serafini 2010]. A strict requirement is that the domains of the ontologies that are being combined are disjoint. This means that the classes/concepts of both ontologies must be separate for ontology linking to be applied. For example, ontology linking is appropriate for aligning disease and symptom ontologies as diseases and symptoms are separate concepts. In the case of disease diagnosis, in order to link symptoms to diseases, combining the SYMP & DOID ontologies is necessary. In principle, the first step of ontology linking is combining the ontologies. To do this, the RDF elements (including all URLs) of both ontologies would need to be combined into a single OWL file. Unfortunately, the lack of copy-paste tools for ontologies makes this impossible. Simple ontology editors such as Protégé do not provide functionalities that would allow RDF elements to be dragged from one ontology into another. Therefore, using the available ontology editing tools to link ontologies is done by adding the elements of one ontology into the other while replicating the elements exactly so not to lose integrity and standard of the ontologies in the process. For the DOID & SYMP ontologies, it is logical to add the symptoms elements from SYMP into the DOID ontology because the DOID ontology is much larger. To do this on a full scale means adding all of the elements of the symptoms ontology into DOID, which is a massive undertaking. However, the process is simple & repetitive and can be applied to connect all diseases classes to their symptoms class. It may even be possible to semi-automate or automate this process. The following is our proposed algorithm of this process:

```

Module linkAlignment (SYMP, DOID) {
  for all diseases D ∈ DOID {
    copy D into DSO
    fetch symptoms of D from a health website or server, a database, etc
    for all symptoms names S of D {
      for symptoms S1 ∈ SYMP {
        if(S == S1.name) {
          1. copy S1 into revised DOID under a base symptoms class
          2. define a new has_symptom object property for D where D
             has_symptom S1
        }
      }
    }
  }
  output DSO
}

```

Listing 2.1: Algorithm for Linking Disease and Symptom Ontologies

Using the above algorithm, the ontology symptoms of a few chosen diseases can be easily aligned with DOID forming a new ontology that is called **Diseases Symptoms Ontology (DSO)**. The DSO includes all the diseases in DOID, and also the symptoms of 11 interrelated diseases. These diseases are *Diabetes (type 1, type 2), Hypertension, Asthma, Adult Respiratory Distress Syndrome, Anemia, Calcemia, Renal Failure, Urinary Tract Infection*. These diseases share some common symptoms are diagnosed on the basis of a certain type of blood or urine test. Figure 2.6 shows the first step in ontology alignment, which is as mentioned earlier combining the SYMP & DOID ontologies.

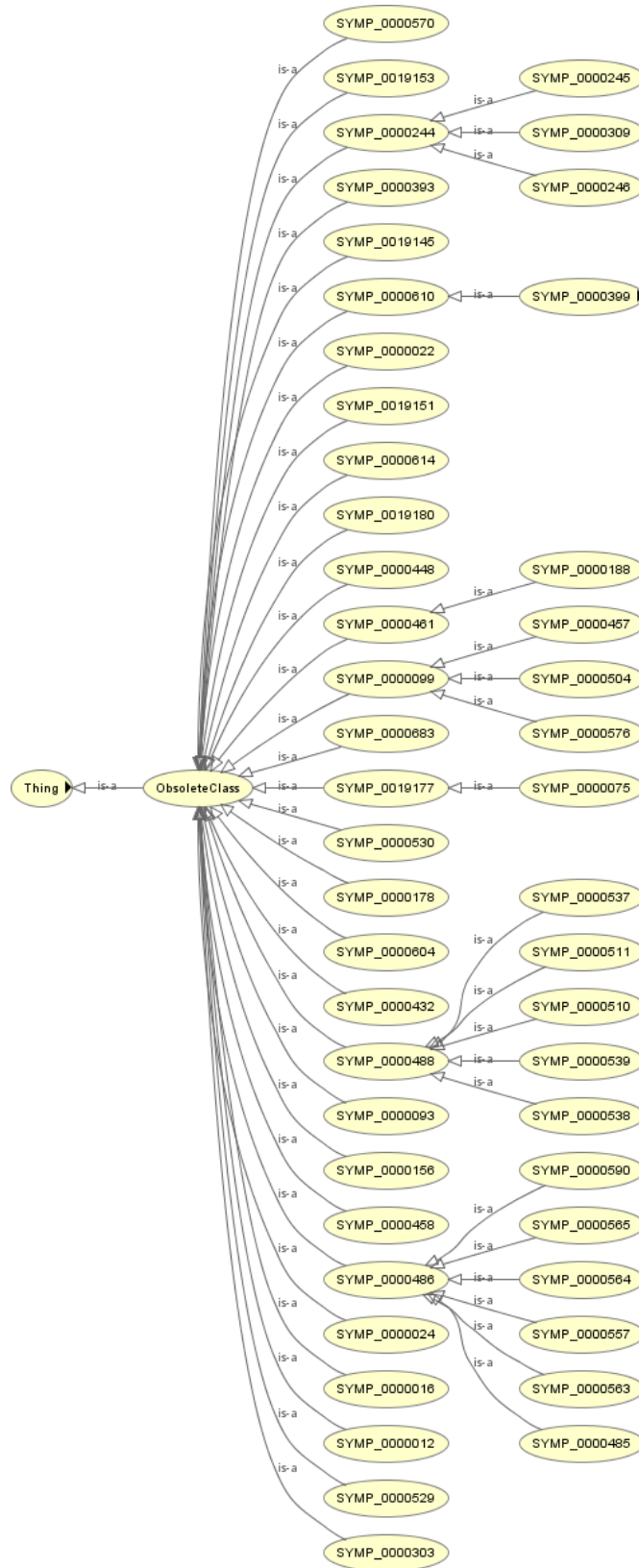


Fig. 2.6: Hierarchy of Symptoms in DSO

Disease and Symptoms classes have annotations that contain vital information such as disease/symptom name and definition. See figure 2.7.

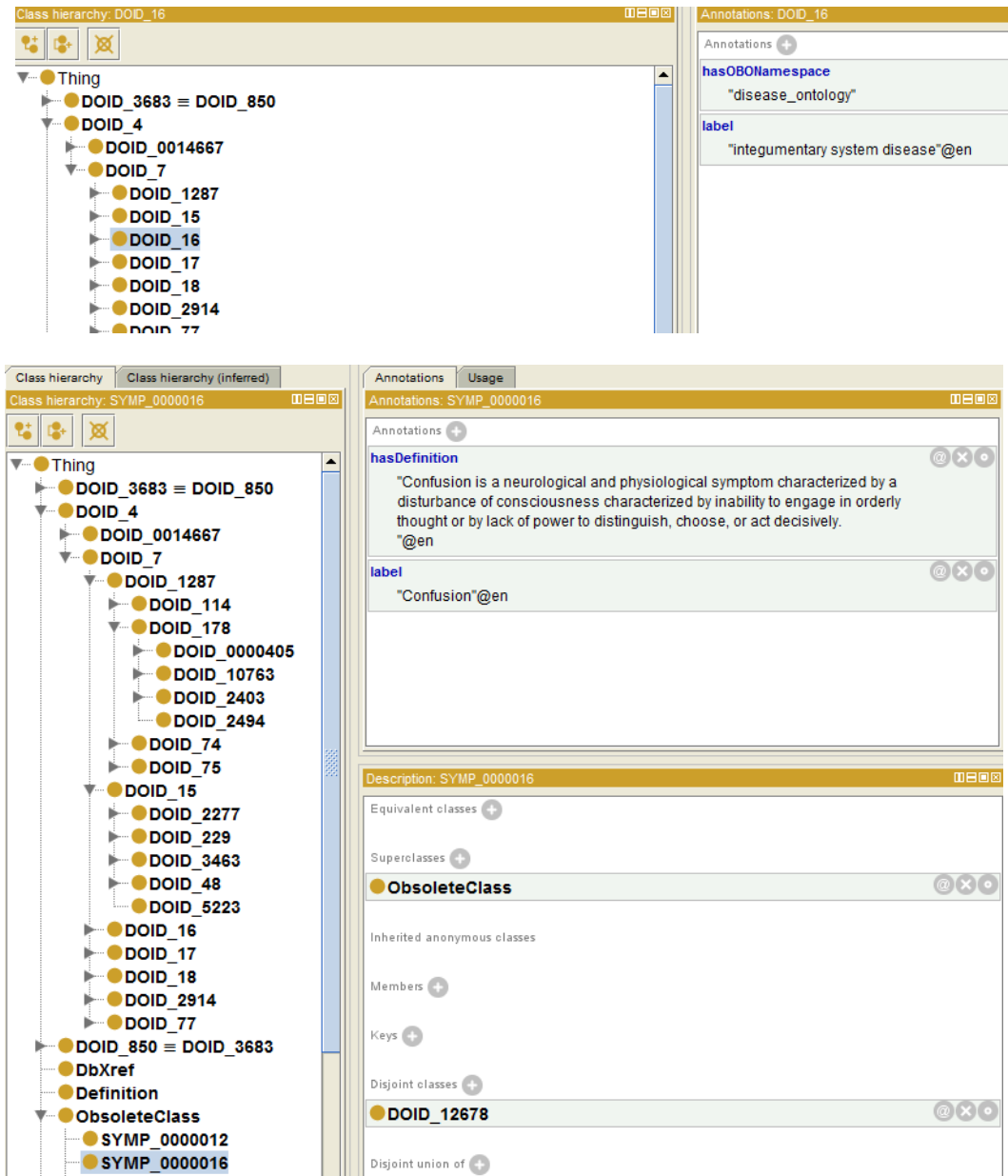


Fig. 2.7: DSO Diseases and Symptoms Class Annotations

Now that the concepts/classes of separate ontologies are combined into a single ontology, it is important to establish relationships between the concepts of the ontologies. In the case of this research, this means establishing relations between the symptoms and diseases class. The most obvious relation is the has_symptom object property. A certain diseases class can be tied to several symptoms classes via several has_symptom properties. Table 2.3 gives is an example of this.

Table 2.3: Hypertension Disease Class (from DOID) linked to Symptoms Classes (from SYMP) representing the Symptoms of Hypertension

Disease Class Name	Disease Class DOID Ontology Code	Object Property	Symptom Class	Symptom Class SYMP Ontology Code
Hypertension	DOID_10763	has_symptom	blurred vision	SYMP_000012
Hypertension	DOID_10763	has_symptom	drowsiness	SYMP_000024
Hypertension	DOID_10763	has_symptom	tinnitus	SYMP_0000393
Hypertension	DOID_10763	has_symptom	nosebleed	SYMP_0000448
Hypertension	DOID_10763	has_symptom	headache	SYMP_0000504
Hypertension	DOID_10763	has_symptom	flushing	SYMP_0000511
Hypertension	DOID_10763	has_symptom	Nausea	SYMP_0000458
Hypertension	DOID_10763	has_symptom	palpitation	SYMP_0000530
Hypertension	DOID_10763	has_symptom	frequent urination	SYMP_0000563
Hypertension	DOID_10763	has_symptom	urgency of urination	SYMP_0000590
Hypertension	DOID_10763	has_symptom	nocturia	SYMP_0000564
Hypertension	DOID_10763	has_symptom	dizziness	SYMP_0000610
Hypertension	DOID_10763	has_symptom	breathing difficulty	SYMP_0019153
Hypertension	DOID_10763	has_symptom	fatigue	SYMP_0019177

Using the protégé editor, this is simple to accomplish. As shown in figure 2.8 below, the above has_symptom properties can be added under the superclasses tab. For each symptom class related to the hypertension disease class, a new has_symptom object property need to be created.

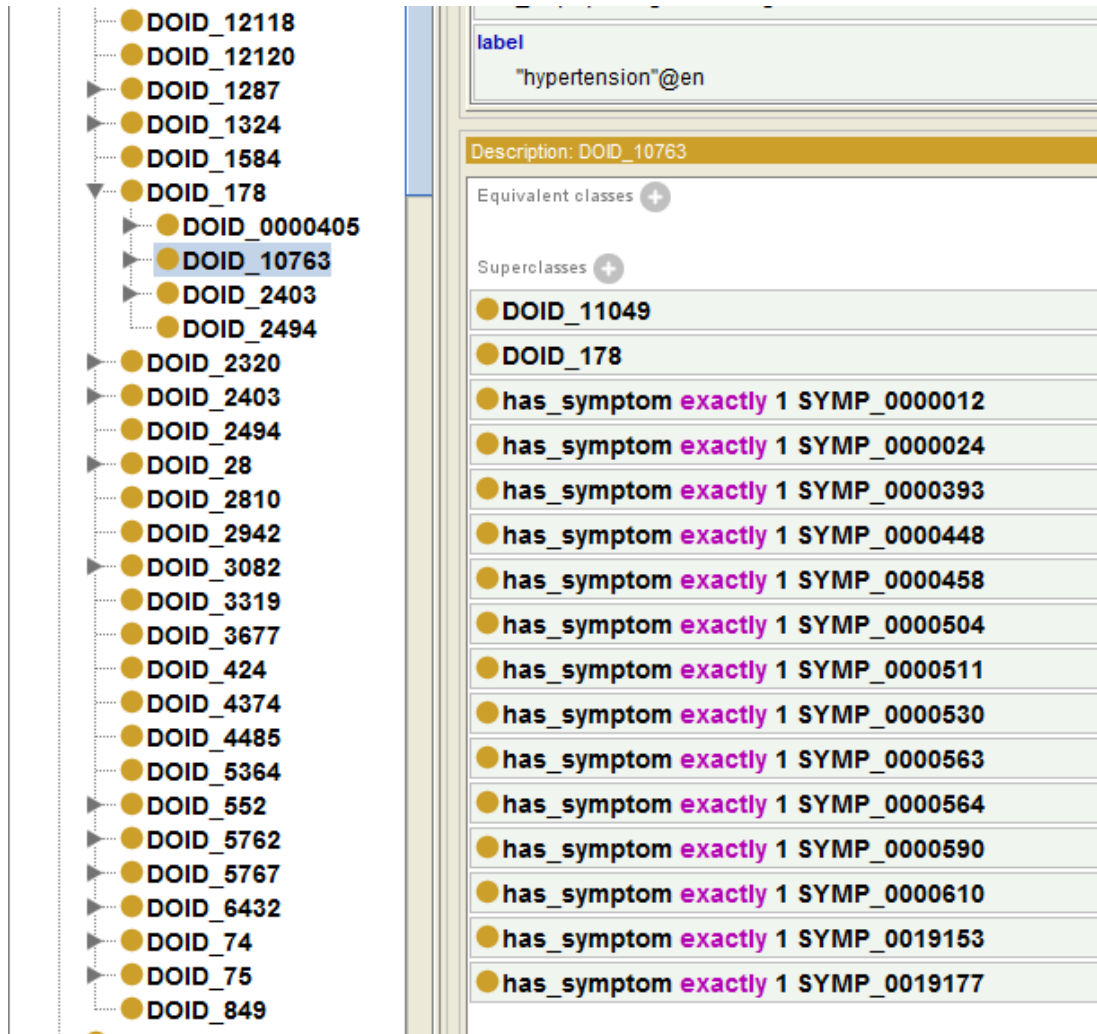


Fig. 2.8: DSO has_symptom property relates a Disease Class to its Symptom Classes

The next two figures are an illustration of how the has_symptom property connects diseases to symptoms. Figure 2.9 shows connections between several DOID diseases and SYMP symptoms, while figure 2.10 shows the connections between the hypertension disease classes to the corresponding symptoms classes in SYMP.

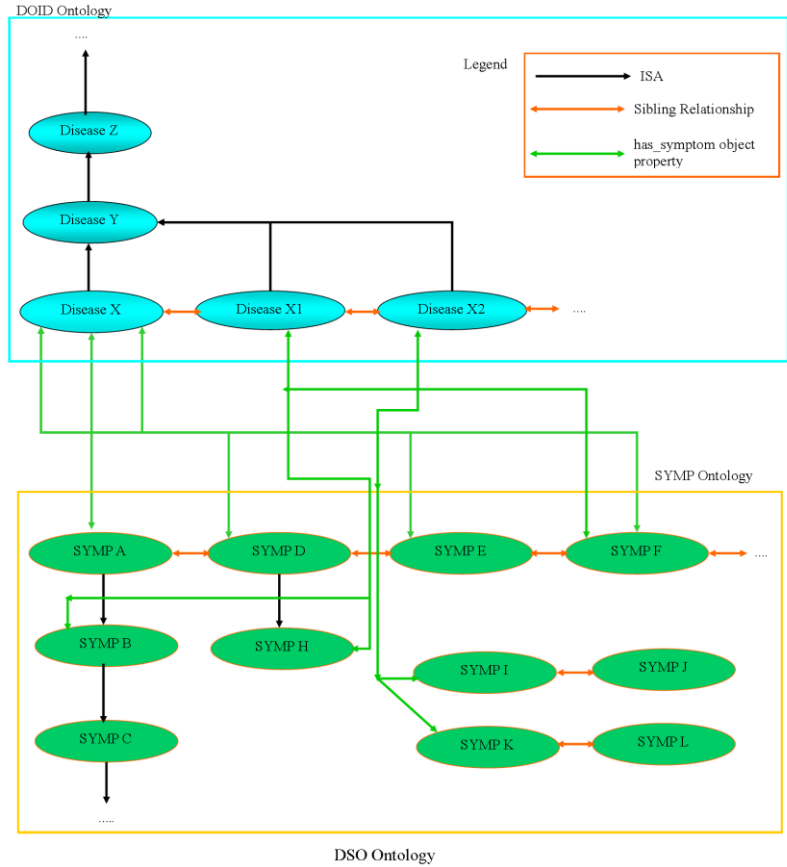


Fig. 2.9: DSO connects SYMP & DOID terms via the has_symptom Property

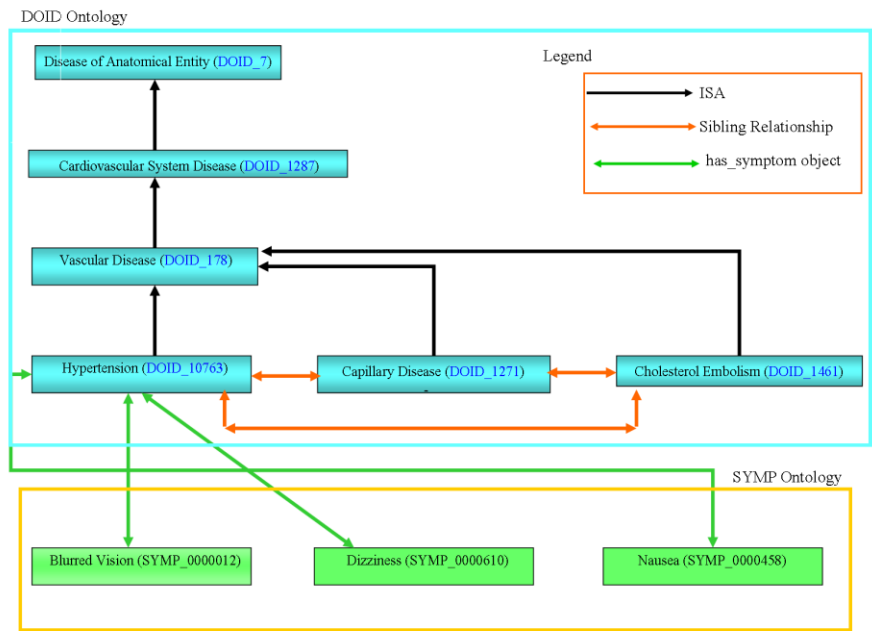


Fig. 2.10: Hypertension DOID disease has_symptom Connections to some of its symptoms in SYMP Ontology

In this chapter we introduced our method to engineer an ontology for differential disease diagnosis. A careful survey of the current research on disease diagnosis ontologies reveals that no effective ontologies are available for general disease diagnosis. Specifically, there is yet no ontology relating diseases to their symptoms. However, our careful investigation also revealed the availability of two OBO standard ontologies that can be used for general disease diagnosis by relating diseases to their symptoms: DOID and SYMP. Section 2.4 described a detailed method and a general algorithm that can be used for linking DOID and SYMP ontologies. The proposed method and process has been repeated for 11 inter-related diseases to produce an annotated and aligned new ontology that we call Diseases Symptoms Ontology (DSO). Indeed our method and process can be repeated for any number of diseases to create a larger, more complete version of the DSO. Our new DSO ontology has been published on our university Flash server (<http://flash.lakeheadu.ca/~omohamme/DSO.owl>) and can be used for analysis by other semantic web applications. In the next chapter, we will design an ontology crawler that can query DSO for specific differential diagnosis information that is of interest to clinicians.

Chapter 3: Developing a DSO OWL Ontology Crawler for Disease Diagnosis

3.1. Ontology Management: Toward Filtering Relational Information

Managing ontologies and annotated data throughout their life-cycles is at the core of semantic systems of all kinds [Hepp *et al.* 2008]. Ontology management infrastructures are needed for the increasing development of semantic applications especially in the corporate semantic web, which comprises the application of semantic technologies in an enterprise environment [Bloehdorn *et al.* 2009]. This is due to the fact that ontologies need to be properly accessed and maintained for ontology-based systems to remain usable. There are two main research approaches in the domain of ontology management. The first considers ontology management as pure steps of ontology accessibility and change performed by the programmer [Klein 2004], while the second takes into account dynamically updating the ontology through extensive learning and evolution management functionalities [Bloehdorn *et al.* 2006]. The first approach is a fundamental management approach required by every semantic web application. On the other hand, the second approach is considered as a secondary approach to ontology management where it is only required for ontology evolution and maturing [Braun *et al.* 2007]. This chapter is concerned with the first approach. It aims to develop an ontology mediator or a manager that handles the ontology accessibility for effective knowledge management through querying, filtering and searching. Other higher level ontology management functionalities such as consistency checking and more inferential primitives are left to our next chapter. Our ontology manager is therefore called ontology crawler since it has a generic engine to filter information from a given ontology (DSO) in a relational way. The term crawler is chosen to describe higher cognitive searching activities by which people determine where places and things are, how to get to them, and actually retrieves them. Crawling is a complex process compared to more simple searching techniques. Our crawling process is designed to mimic the way doctors perform differential diagnosis. Doctors attempt to ask themselves question to know what is happening with their patient then go and test their ideas while keeping other options open. This means knowing diseases and conditions, their signs and symptoms, and conducting an investigation (including tests) to rule things in or out. The type of questions required for differential diagnosis represent relations between diseases and symptoms in a variety of formations. Examples of such questions are:

- What are the symptoms of a given disease?
- If a patient displays a number of symptoms, then what are the possible diseases he/she may have?
- If a patient displays a number of symptoms, and a certain disease is suspected then what symptoms of the disease are not displayed by the patient (the so called missing symptoms)?

- What diseases are related to each symptom displayed by the patient?

The crawling process should be driven by these questions to aid doctors with the process of diagnosis and narrowing their choices. The other motivation behind designing our own ontology crawling engine is based on the fact that the available semantic web engines like Swoogle³², OntoSearch³³ and OntoKhoj [Patel *et al.* 2003] allow ontologies to be searched using only keywords (e.g. classes), but further refinement of the search criteria based on the relational structure of the ontology is not possible. Our developed ontology crawler filters relevant information subject to the relational structure of the given ontology. The developed ontology crawler is designed to search for properties associated with classes in variety of relational formations. Our ontology crawler provides higher applications with primitives for ontology navigation which make the task of finding information more effective, efficient, and interactive.

3.2. The Basic Infrastructure for Programming the Ontology Crawler

The infrastructure that we are describing in this section uses a popular open source Semantic Framework: Jena³⁴. Jena is a Java framework for building Semantic Web applications. Jena provides a programmatic environment for RDF, RDFS and OWL, including a rule-based inference engine (see Figure 3.1). Jena uses SPARQL to query the ontology. However, the use of SPARQL query engine is restrictive for the purposes of the hierarchical relational navigation required for our ontology crawler. Also, in order to write SPARQL queries, the programmer must manually match the exact RDF/OWL ontology structure for the prescribed query [Polleres *et al.* 2009]. This process is restrictive and low level as it requires exact pattern matching expertise for aligning the query to the specified ontology. By using Jena OWL primitives, the programmer is relieved from this pattern matching. These primitives can be used to retrieve information including class labels (for example disease names and synonyms), object and data type properties from an ontology class. This is one of the reasons that motivated our work to develop higher level of ontology querying module using Jena. The next section describes the functionalities of our OWL ontology crawler.

³² <http://swoogle.umbc.edu/>

³³ <http://www.ontosearch.org/>

³⁴ <http://jena.sourceforge.net/>

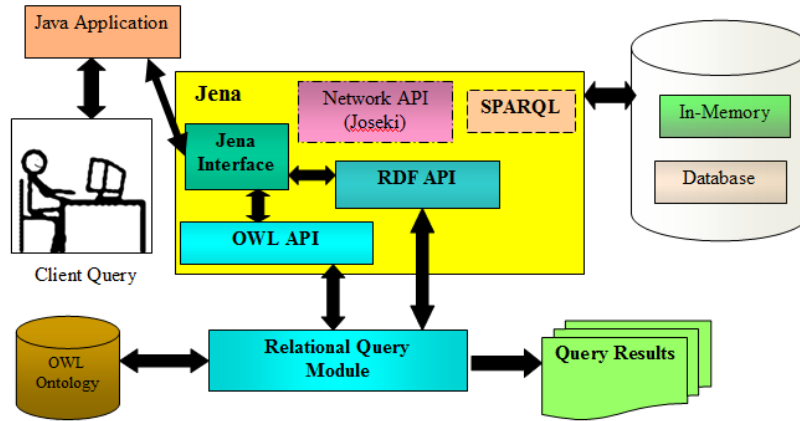


Fig. 3.1: Jena Ontology Query Architecture

3.3. Developing the DSO Crawling Relational Primitives

In this section we are describing our DSO ontology crawler software architecture consisting of Jena components along with the required relations for combining the ontological attributes in a meaningful way for the purpose of inferring new knowledge necessary for the differential diagnosis process. The ontology crawler replaces the Jena SPARQL querying engine by implementing the necessary differential diagnosis relations using Jena OWL API primitives. Our ontology crawler is part of an overall architecture that aims to serve health care providers with the notion of a differential diagnosis recommendation system. Figure 3.2 illustrates an overview of our overall architecture where the ontology crawler is one of its major components.

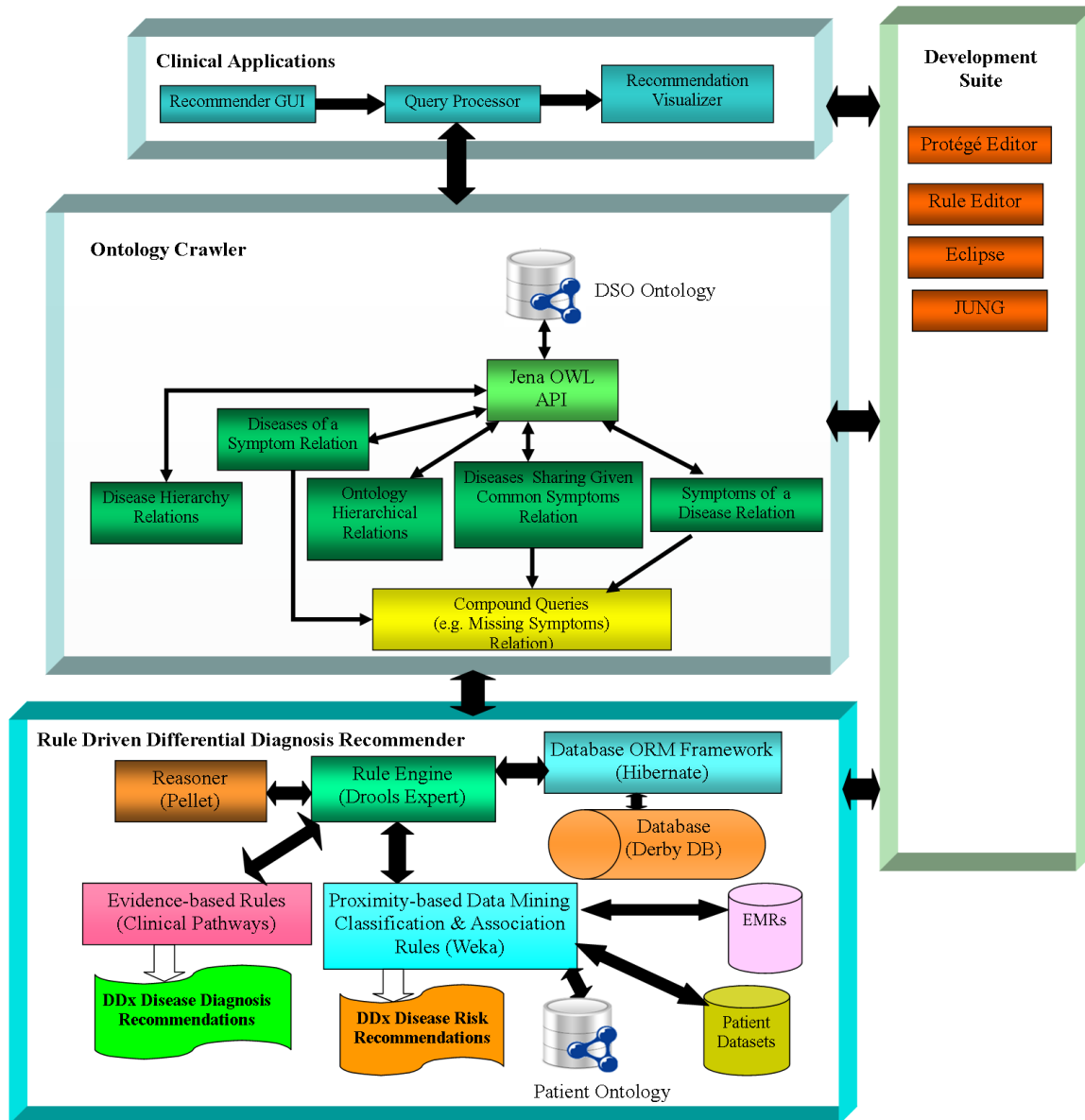


Fig. 3.2: An Overview of the Semantic Web Differential Diagnosis Recommendation Engine Model

The **ontology crawler** is primarily used in our differential diagnosis engine for **retrieving** information from the DSO ontology using representative relations. The representative relations address common queries that health care professionals use during the process of differential diagnosis. Figure 3.3 provides the most important representative relations that we employed within our ontology crawler along with their formal representation.

1. What are the diseases for a given symptom?
R1: $S[1] \rightarrow D[N]$
2. What are the diseases sharing number of common symptoms?
R2: $S[M_1] \rightarrow D[M_2]$
3. What are the symptoms for a given disease?
R3: $D[1] \rightarrow S[A]$
4. What is the ontological knowledge structure for a given disease?
(i.e. Disease Hierarchy)?
R4: $D[1] \rightarrow D_{tree} [D_{os1}, D_{os2}, D_{os3}, \dots, D_{osn}, D_{oc1}, D_{oc2}, D_{oc3}, \dots, D_{ocn}]$
5. What are the missing symptoms for a given disease?
R5: $S[B] \rightarrow D[A] \rightarrow D_i \in D[A] \rightarrow S[D_i] - S[B]$
6. What are the diseases for a given pathology where they share number of common symptoms?
R6: $S[M_1], P_i \rightarrow D[M_2] \in P_i$

Fig. 3.3: DSO Ontology Crawler's Six Representative Relations

Each of the representative relations (R1 to R6) interact with the DSO ontology where their domain represents the inputs and their range represents the outputs. The relation's domains/ranges include the set of the DSO ontology attributes including disease, diseases, symptom, symptoms and pathology. Figure 3.4 illustrates each relation targeted domain and range.

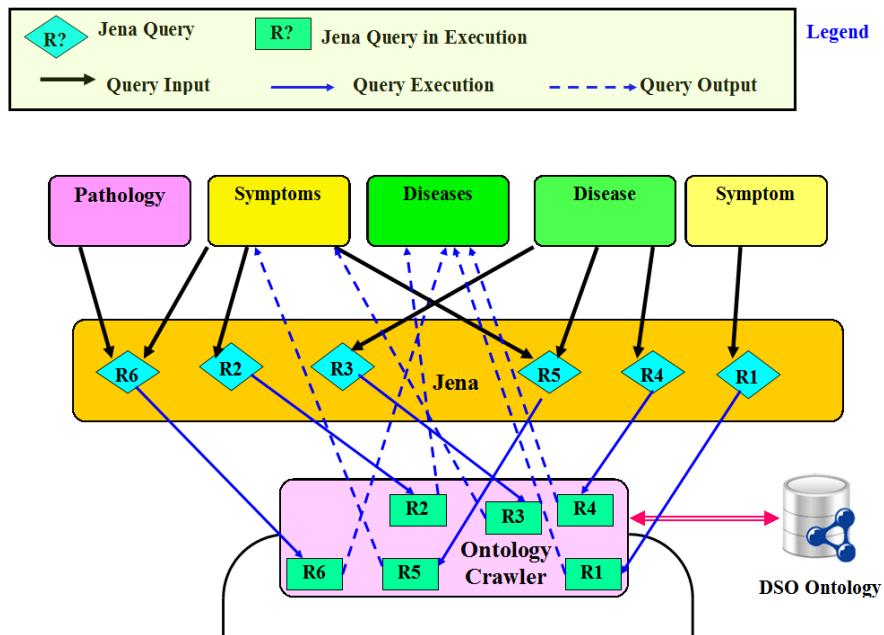


Fig. 3.4: The Ontology Crawler's Six Representative Relations

In the following sections we describe each relation in details.

3.3.1 The R1 Relation Description

The R1 relation takes a symptom as input and outputs DSO diseases that display that symptom. It is designed to answer the question in diagnosis which is: If a patient displays a certain symptom, then what diseases may he/she have? Figure 3.5 is a graphical representation of R1. R1 is a special case of R2. R1 takes a single symptom as input, but R2 can take more than one symptom.

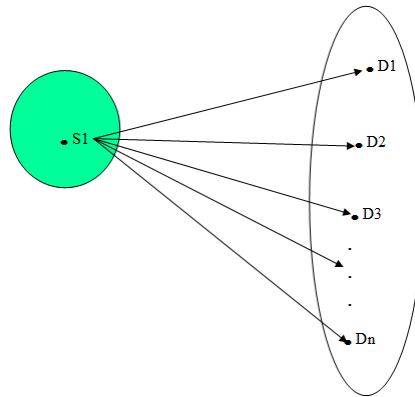


Fig. 3.5: R1 Relation

3.3.2 The R2 Relation Description

The R2 relation is designed to answer a fundamental question in the differential diagnosis process. It replies to question: If a patient displays a number of symptoms, then what are the possible diseases the patient may have? The layout of R2 is shown in figure 3.6.

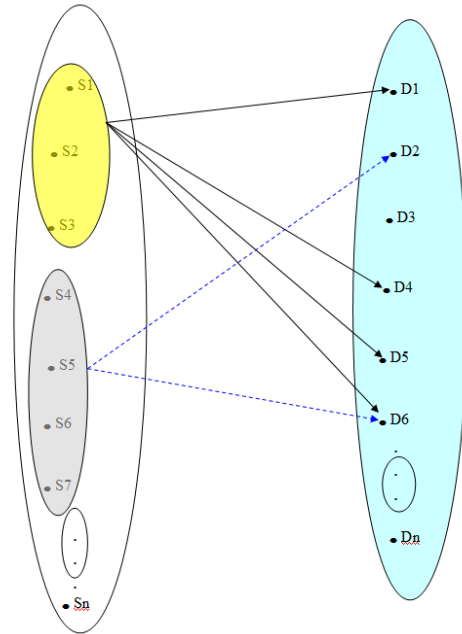


Fig. 3.6: R2 Relation

The pseudo code for R2 is shown below.

Module R2(Symptoms S[M₁], OntModel m) {

// R2: Find diseases sharing number of common symptoms

diseaseList[][] = null;

counter = 0;

diseases[] = null;

returnedDiseases[] = null;

begin

searchForDiseasesWithSymptoms(S[M₁], m)

returnedDiseases[] = diseaseList[0][] ∩ diseaseList[1][] ∩ diseaseList[2][] ∩ diseaseList[3]

∩ ∩ diseaseList[M₁ - 1];

output returnedDiseases[]

end

}

void searchForDiseasesWithSymptoms(Symptom[] S1, OntModel m) {

for (int i = 0; i < S1.size; i++) {

searchForDiseasesWithSymptom(S1[i], m);

}

}

void searchForDiseasesWithSymptom(Symptom S, OntModel m) {

for each OntClass c in OntModel m {

checkDiseaseClassForSymptom(c, S);

```

    }
    diseaseList[counter][] = diseases;
    counter++;
    diseases = null;
}
void checkDiseaseClassForSymptom(OntClass c, Symptom S) {
    List<OntClass> disjointClasses = c.listDisjointWith();
    String symptom = null;
    List<RDFNode> disjointLabel = null;
    while(disjointClasses.hasNext()) {
        disjointLabel = disjoint.next().listLabels(LANG);
        while (disjointLabel.hasNext()) {
            symptom =
            disjointLabel.next().asLiteral().toString() ;
            if(symptom.equalsIgnoreCase(symptomName))
                diseases.add(c);
        }
    }
}
}

```

Figure 3.7 shows one execution of R2. Here, the input to R2 is the symptoms Nausea and Fatigue. The output is some DSO diseases that have Fatigue and Nausea among their symptoms. These diseases are listed below:

```

diabetes mellitus type 2
diabetic ketoacidosis
lipoatrophic diabetes mellitus
diabetic peripheral angiopathy
hypertension
renal failure
hypercalcemia

```



```

13
14  /**
15   * @param args
16   */
17  public static void main(String[] args) {
18      // TODO Auto-generated method stub
19
20      try{
21          OntModel m = ModelFactory.createOntologyModel( OntModelSpec.OWL_DL_MEM);
22          m.getDocumentManager().addAltEntry(null, "c:/ActionRecognition/Ontology/DSO.owl");
23          m.getDocumentManager().addAltEntry(null, "c:/ActionRecognition/Ontology/DSO.owl" );
24
25          m.read("file:/C:/ActionRecognition/Ontology/DSO.owl");
26
27          List<String> symp = new LinkedList<String>();
28          //symp.add("Vomiting");
29          //symp.add("shortness of breath");
30          //symp.add("urinary frequency");
31          symp.add("fatigue");
32          symp.add("nausea");
33
34          OntologyClassQuery q = new OntologyClassQuery();
35          List<String> query = q.linkSymptomsToDiseases(System.out, m, null, symp);
36          //List<String> query = q.getSymptomsOfDisease(System.out, m, "hypertension");
37          //List<String> query = q.getMissingSymptoms(System.out, m, symp);
38          //List<String> query = q.linkSymptomsToDiseasesOfPathology(System.out, m, symp, "glucose metabolism disease");
39          //new OntologyClassQuery().showHierarchy(System.out, m);
40          for(int i = 0; i < query.size(); i++) {
41
42              System.out.println(query.get(i));
43          }
44
45      }
46
47      catch(Exception e) {
48          e.printStackTrace();
49      }
50
51  }

```

```

<terminated>- Main (1) [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (2011-07-24 1:18:19 AM)
diabetes mellitus type 2
diabetic ketoacidosis
lipoatrophic diabetes mellitus
diabetic peripheral angiopathy
hypertension
renal failure
hypercalcaemia

```

Fig. 3.7: Executing R2 to Find the Diseases that Display Nausea and Fatigue as Symptoms

Figure 3.8 illustrates a UML sequence diagram for the dynamics of executing R2 for an input scenario used in figure 3.7.

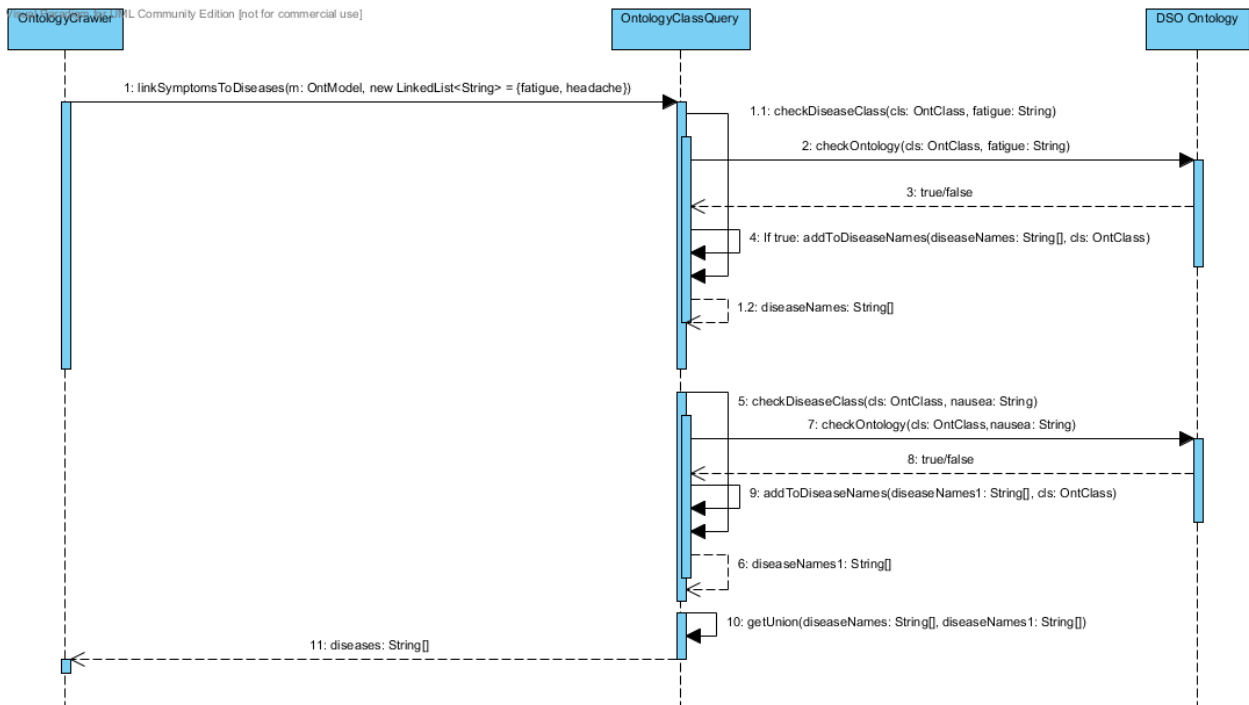


Fig. 3.8: Sequence Diagram for R2

3.3.3 The R3 Relation Description

The R3 relation answers a basic question in differential diagnosis: What are the symptoms of a given disease? Figure 3.9 is a graphical representation of R3.

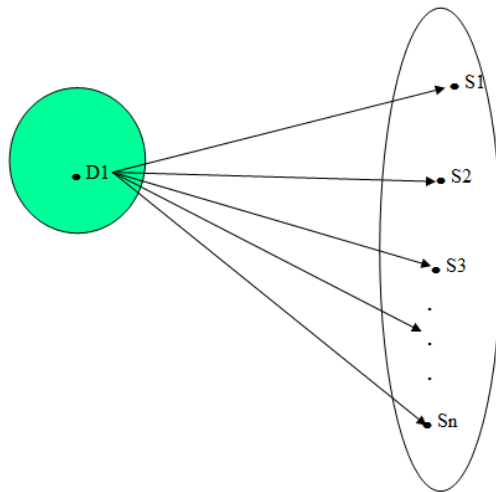


Fig. 3.9: R3 Relation

The pseudo code for R3 is as follows:

```

Module R3(Disease D, OntModel m) {
  symp[] = null;
  Iterator<OntClass> clsDisjoint;
  begin
    for each OntClass c in OntModel m {
      if(c.getLabel() == D) {
        clsDisjoint = c.listDisjointWith();
        while(clsDisjoint.hasNext()) {
          sympt = clsDisjoint.next().getLabel(LANG);
          if(!symp.contains(sympt)) {
            symp.add(sympt);
          }
        }
      }
    }
  end
}

```

The figure below (figure 3.10) is the sequence diagram for R3.

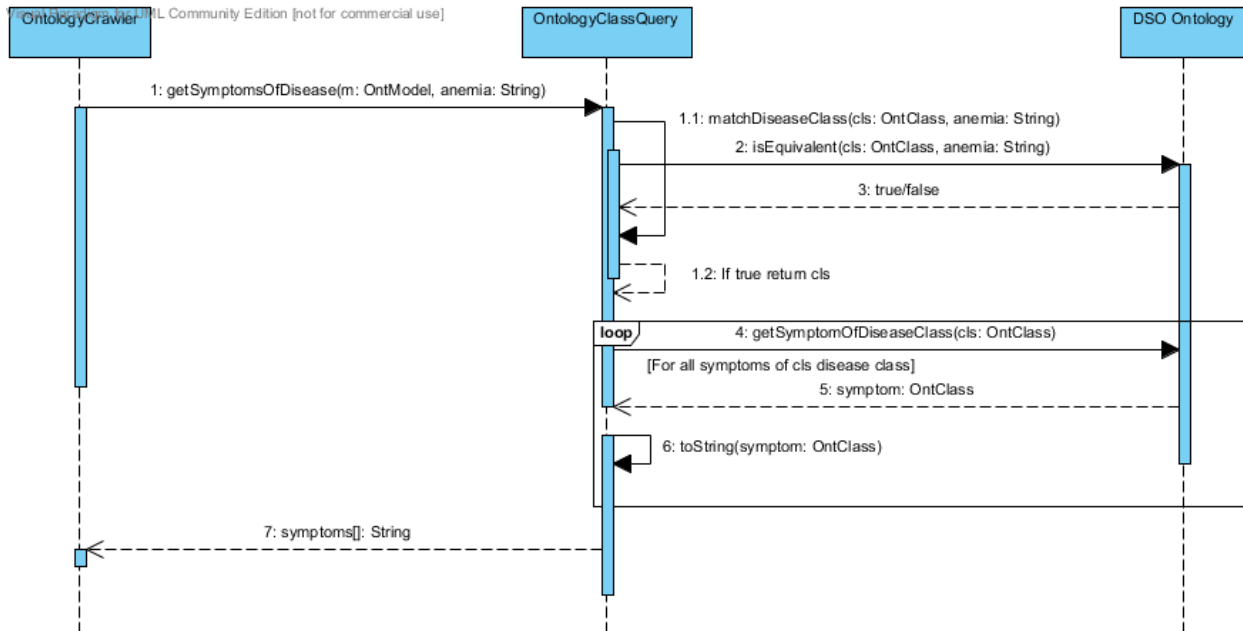


Fig. 3.10: R3 Sequence Diagram

One example input to R3 is the disease anemia. The output of R3 in this case would be the symptoms of anemia. These symptoms are listed below:

fatigue
 headache
 shortness of breath
 chest pain
 abnormal heart beats
 paleness
 dizzy

3.3.4 The R4 Relation Description

The R4 relation displays the hierarchy of a disease showing the pathology classification it belongs to as well as its sibling diseases. Figure 3.11 is a graphical representation of R4.

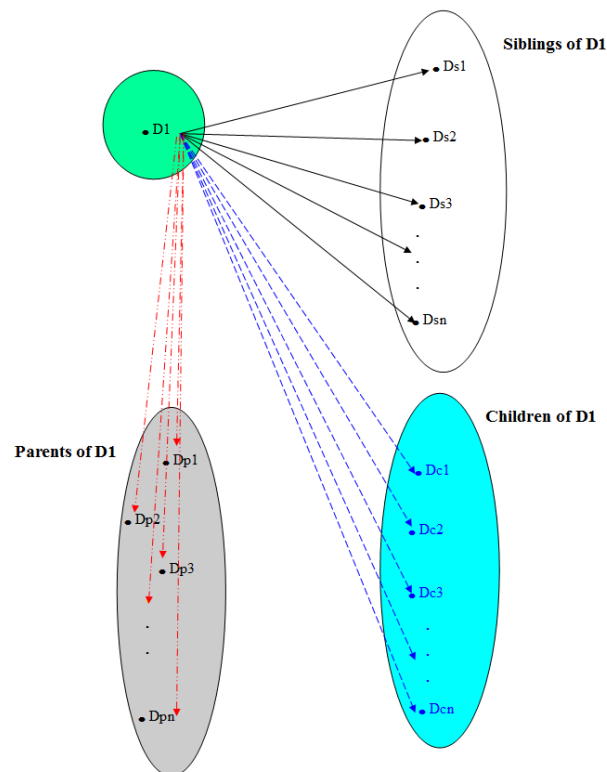


Fig. 3.11: R4 Relation

The output below shows the disease hierarchy of type 2 diabetes (diabetes mellitus type 2).

```
Class :DOID#DOID_0050013 carbohydrate metabolism disease
Class :DOID#DOID_4194 glucose metabolism disease
Class :DOID#DOID_9993 hypoglycaemia
Class :DOID#DOID_9351 diabetes mellitus
Class :DOID#DOID_9744 diabetes mellitus type 1
Class :DOID#DOID_9352 diabetes mellitus type 2
```

```

Class :DOID#DOID_1837 diabetic ketoacidosis
Class :DOID#DOID_11712 lipoatrophic diabetes mellitus
Class :DOID#DOID_10182 diabetic peripheral angiopathy
Class :DOID#DOID_11717 neonatal diabetes mellitus
Class :DOID#DOID_11716 prediabetes syndrome
Class :DOID#DOID_11714 gestational diabetes
Class :DOID#DOID_4195 hyperglycemia

```

3.3.5 The R5 Relation Description

Like R2, the R5 relation takes in a number of symptoms as input and finds the diseases that display the inputted symptoms. However, R5 goes further than R2 by asking the user to select one of these diseases. For the selected disease, it displays the missing symptoms. The missing symptoms of are symptoms of the selected diseases that were not given as input to R5. R5 gets the missing symptoms by using R3 to get the symptoms of the disease and subtracting the input symptoms from the symptoms of the disease. Figure 3.12 is a graphical representation of R5.

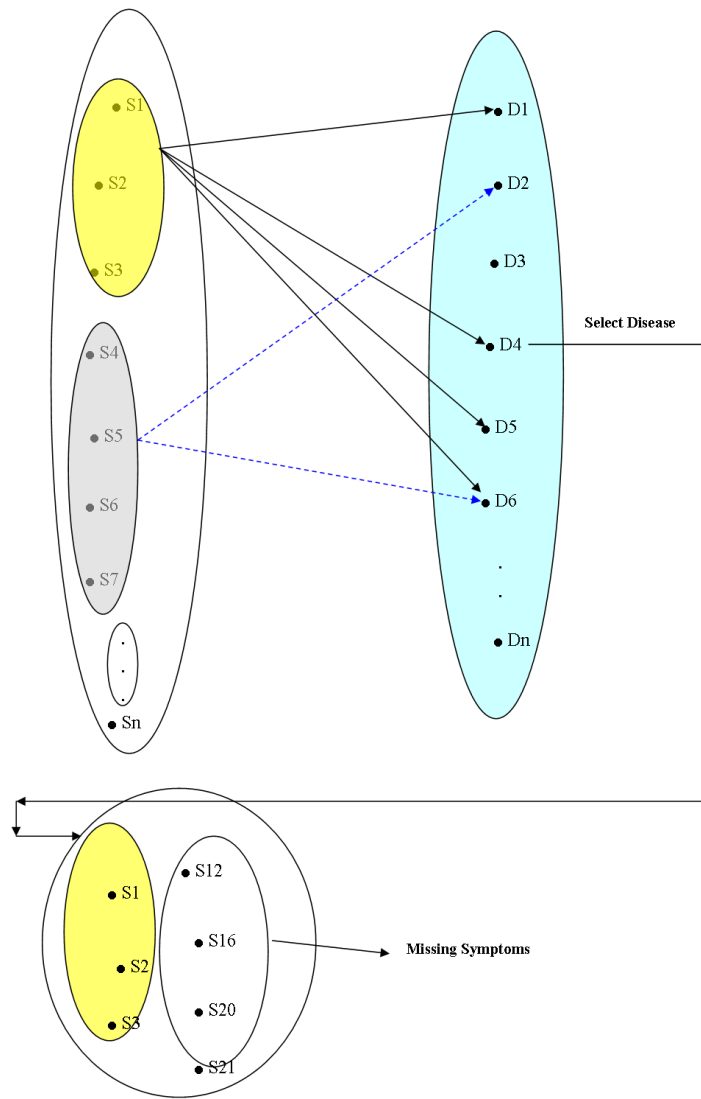
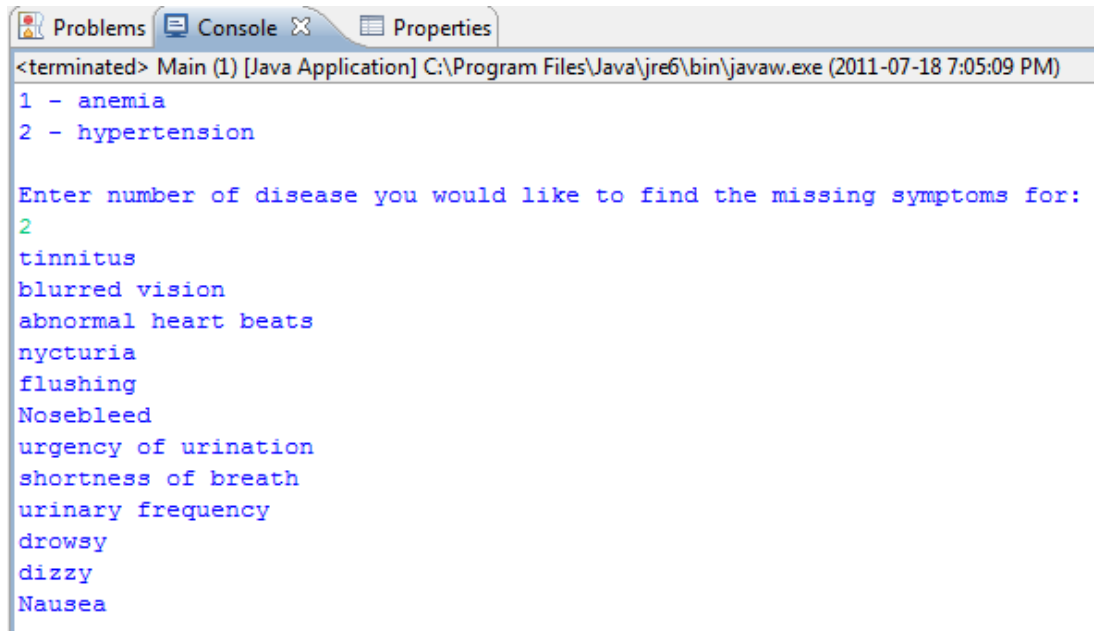


Fig. 3.12: R5 Relation

One example input to R5 would be the symptoms Fatigue and Nausea. R5 would use R2 to link the two symptoms to the diseases that display them. Then R2 would ask the user to choose one of the diseases. Then R2 would display the missing symptoms for the selected disease. Figure 3.13 illustrates the above example execution of R5. Figure 3.13 shows the diseases the display the symptoms fatigue and headache. It then shows the user selecting one the diseases (hypertension). After the user selects a disease, the missing symptoms of the disease are displayed.



```
<terminated> Main (1) [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (2011-07-18 7:05:09 PM)
1 - anemia
2 - hypertension

Enter number of disease you would like to find the missing symptoms for:
2
tinnitus
blurred vision
abnormal heart beats
nycturia
flushing
Nosebleed
urgency of urination
shortness of breath
urinary frequency
drowsy
dizzy
Nausea
```

Fig. 3.13: Example R5 Result

3.3.6 The R6 Relation Description

The R6 relation is similar to R2. Like R2, it finds diseases that share a number of given symptoms. However, R2 searches the entire DSO for diseases that share the given symptoms while R6 searches specified disease pathology only. Therefore, R6 would take a number of symptoms and a disease pathology as input and output diseases that share the inputted symptoms and also belong to the given disease pathology. Figure 3.14 is an illustration of R6.

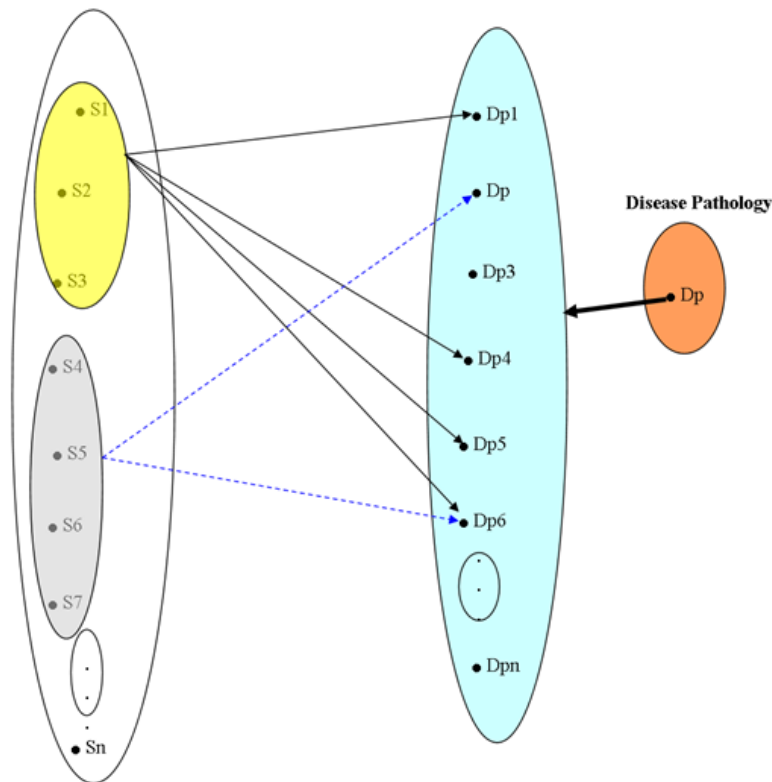


Fig. 3.14: R6 Relation

One example R6 execution would be to find the diseases that are considered to be vascular diseases and display the Nausea and Fatigue symptoms. The results of this R6 example are as follows: [hypertension](#)

3.4. Enhancing the Power of the Ontology Crawler: The Integration of Pellet Reasoner

In order to enhance the power of the six relationships described above, a variety of ontology-based reasoners (e.g. Pellet³⁵, Fact++³⁶, KAON2³⁷) should be used to derive additional truths about concepts from the provided ontology model. For our purpose we used Pellet because it is a complete OWL 2 DL reasoner implemented in Java and it is open source. It also features some rule-based capability (e.g. SWRL).

The Pellet inference engine adds power to the Ontology Crawler. For example when executing R2 with a Jena-only ontology crawler, solely the diseases in DSO where the given symptoms are

³⁵ <http://clarkparsia.com/pellet>

³⁶ <http://owl.man.ac.uk/factplusplus/>

³⁷ <http://kaon2.semanticweb.org/>

defined are returned. This is called direct inference. This is an approach that does not make use of the parent-child relationships of DSO. When adding a reasoner however, it can infer further findings through the parent-child relationships. For R2 for example, the reasoner makes the following conclusion: If the given symptoms are defined for a certain parent disease in DSO, then all its child diseases inherit these symptoms. This is called indirect inference making and is semantically correct, in OWL 2 DL, as all child diseases' classes are subtypes of a base disease class and inherit its symptoms, and in fact all of its attributes. From a clinician's point of view, the additional inference making provides further diagnosis options that can be considered especially when the options obtained from direct inference are ruled out.

Jena includes support for a variety of reasoners through the inference API. A common feature of Jena reasoners is that they create a new RDF model, which appears to contain the triples that are derived from reasoning, as well as the triples that were asserted in the base model. This extended model still conforms to the contract for Jena models. So it can be used wherever a base model can be used. The ontology API exploits this feature: the convenience methods the ontology API provides can query an extended inference model in just the same way as a plain RDF model. There are two different ways to use Pellet in a Jena program: either using a direct Pellet interface (common approach); or using Jena DIG interface (less common approach). The direct Pellet interface is much more efficient (e.g. does not have the HTTP communication overhead) and provides more inferences (DIG protocol has some limitations). Based on the direct interface, we can easily integrate pellet into our Jena Ontology Crawler as follows:

```
OntModel ontModel = ModelFactory.createOntologyModel( PelletReasonerFactory.THE_SPEC);
ontModel.getDocumentManager().addAltEntry(null, "c:/ActionRecognition/Ontology/DSO.owl");
ontModel.getDocumentManager().addAltEntry(null, "c:/ActionRecognition/Ontology/DSO.owl" );
ontModel.read("file:/C:/ActionRecognition/Ontology/DSO.owl");
// get the underlying Pellet graph
PelletInference pellet = new PelletInference((PelletInfGraph) ontModel.getGraph());
//progressBar.setIndeterminate(true);
boolean consistent = pellet.isConsistent();
System.out.println("Consistency = " + consistent);
// Trigger classification
pellet.classify();
// Trigger realization
pellet.realize();
```

In the above code, the new Jena RDF model is created using a PelletInference object, which takes the base Jena RDF model (OntModel m) and converts it to the new pellet-reasoned model. The same object can then be used to check the new model for consistency (pellet.isConsistent()) and add the additional pellet inferences to it (pellet.classify()). Results, with pellet, of R2 using the symptom Nausea are shown below:

```
renal failure
chronic kidney failure
end stage renal failure
uremia
hemolytic-uremic syndrome
uremic neuropathy
```


acute kidney failure
acute kidney tubular necrosis
hepatorenal syndrome
renal hypertension
malignant hypertensive renal disease
renovascular hypertension
benign hypertensive renal disease
HELLP syndrome
hypertension
pre-eclampsia
eclampsia
severe pre-eclampsia
mild pre-eclampsia
malignant hypertension
malignant essential hypertension
malignant secondary hypertension
malignant renovascular hypertension
hypertensive cardiopathy
malignant hypertensive heart disease
benign hypertensive heart disease
secondary hypertension
benign secondary hypertension
benign renovascular hypertension
pulmonary hypertension
persistent fetal circulation syndrome
essential hypertension
benign essential hypertension
diabetes mellitus type 2
diabetic peripheral angiopathy
lipoatrophic diabetes mellitus
diabetic ketoacidosis
hypercalcemia

Results of R2, with pellet reasoning, using the two symptoms fatigue and headache are shown below.

hemolytic-uremic syndrome
renal hypertension
malignant hypertensive renal disease
renovascular hypertension
benign hypertensive renal disease
hemoglobinuria
HELLP syndrome
pancytopenia
Fanconi's anemia
anemia
microcytic anemia
fetal erythroblastosis
hypochromic anemia
congenital anemia
congenital hemolytic anemia
congenital nonspherocytic hemolytic anemia
congenital dyserythropoietic anemia
hereditary spherocytosis
hemoglobinopathy
methemoglobinemia

hemoglobin D disease
hemoglobin E disease
hemoglobin C disease
Diamond-Blackfan anemia
deficiency anemia
protein-deficiency anemia
neonatal anemia
kernicterus due to isoimmunization
twin-to-twin transfusion syndrome
anemia of prematurity
macrocytic anemia
megaloblastic anemia
aplastic anemia
pure red-cell aplasia
sideroblastic anemia
sideroblastic anemia with spinocerebellar ataxia
X-linked sideroblastic anemia
pyridoxine-responsive sideroblastic anemia
pyridoxine-refractory autosomal recessive sideroblastic anemia
congenital hypoplastic anemia
normocytic anemia
hemolytic anemia
hypertension
pre-eclampsia
eclampsia
severe pre-eclampsia
mild pre-eclampsia
malignant hypertension
malignant essential hypertension
malignant secondary hypertension
malignant renovascular hypertension
hypertensive cardiopathy
malignant hypertensive heart disease
benign hypertensive heart disease
secondary hypertension
benign secondary hypertension
benign renovascular hypertension
pulmonary hypertension
persistent fetal circulation syndrome
essential hypertension
benign essential hypertension

Two major achievements have been introduced in this chapter. The first is to formulate the high-level clinical queries related to differential diagnosis investigations into generic programmable relations. This chapter introduces six fundamental relations that link symptoms and diseases for the purpose of differential diagnosis. The second achievement is to integrate the developed relations as a new relational query engine for the Jena framework instead of its traditional SPARQL low-level query engine. This chapter terms the new Jena relational query engine as the DSO Crawler. The crawler is a component of a semantic web based model for differential diagnosis recommendation (Figure 3.2). Finally this chapter demonstrates the design and use of the various developed relational queries using several software engineering forms including screen shots, code snippets, block diagrams and sequence diagrams. These illustrations shed light

on the complexity of the development work behind the DSO crawler as well as the simplicity of using these relations by the clinicians for inference and decision making.

Chapter 4: Rule-based and Proximity-based Differential Diagnosis Recommenders

A health provider's ability to make correct decisions regarding patient care is predicated on the correct identification of a patient's disease. However, the process of developing diagnostic certainty remains a challenging task despite an increasingly sophisticated array of available diagnostic modalities and techniques. Clinicians need support to integrate a broad range of findings from these tools along with a patient's symptoms and signs [Weiner, Pifer, and Williams 2005]. This chapter develops a rule-based recommender as a utility: computer-aided differential diagnostic decision recommender using two new approaches. The first approach uses semantic web technologies based on flexible clinical pathways for guiding clinicians to provide test results for determining a diagnostic decision. This approach is called evidence-based approach. The second approach utilizes semantic web technologies for navigating clinical documents and mining clinical and diagnostic indicators. The second approach is called the proximity-based approach. This chapter demonstrates that the process of disease diagnosis requires both these approaches.

4.1. Disease Diagnosis: Evidence-based vs. Proximity-based

Clinical diagnosis is a process of finding and establishing the characteristics and type of an illness that a person is suffering from based on signs, symptoms and laboratory findings along with some predictive methods. Therefore the clinical diagnosis process is a complex, loosely defined, and it is a multistep process that requires analysis from different perspectives. Generally, clinical diagnosis involves a series of iterative steps [Reddy 2010]:

1. Taking patient's history
2. Physical examination and systemic examination
3. Analyzing the patient's data
4. Differential diagnosis (DDx) to yield provisional diagnosis
5. Further examinations including laboratory examination
6. Confirming or refuting the diagnosis (which requires going back to step 4)
7. Starting the treatment

Traditionally, this iterative process can be guided using medical decision support systems. Legacy medical diagnosis systems, in general, represent some sort of content-based recommendation system which analyzes symptoms, signs, and descriptions to identify diagnoses that are of particular interest to the clinician [Pazzani *et al.* 2007]. Many researchers implemented the content-based diagnostic principles via a decision support system or expert system that provides suggested actions for physicians based on individual patient characteristics

and established treatment protocols. Such systems may enable physicians to make better-quality decisions, and may enable patients to more consistently follow medical recommendations [Diamond 2004]. For this purpose, *rules* have been used to model whatever available knowledge that can help in the diagnosis process. Table 4.1 illustrates some of the notable legacy attempts in this direction. Most proposed systems cover a limited number of diseases and some utilize very restrictive types of diagnostic technologies such as direct deductive systems [Yeh *et al.* 1990], decision trees [Wim *et al.* 2003] or neural networks [Matsumoto *et al.* 2004].

The rule-based representation for clinical diagnosis began gaining research support after the MYCIN [Buchanan and Shortliffe 1984] reported success during the early seventies. MYCIN is the name of a decision support system developed by Stanford University, built to assist physicians in the diagnosis of infectious diseases. The system (also known as an "expert system") would ask a series of questions designed to emulate the thinking of an expert in the field of infectious diseases. From the responses to these questions, the system gives a list of possible diagnoses with a probability attached to each possible diagnosis, as well as recommend treatment (i.e. it provides "decision support")³⁸. The name "MYCIN" actually comes from antibiotics, many of which have the suffix "-mycin". In literature there are many other medical diagnosis systems that use rules (e.g. Easy Diagnosis [Martin 2004], INTERNIST-I [Kumar *et al.* 2009] and ONCOCIN [Wiederhold *et al.* 2001]). The specification of rule-based knowledge is a flexible way for designing medical recommendation systems. Rules are a way of intuitive knowledge representation for building intelligent systems. Additionally, in the context of the semantic web initiative, the definition of a general rule interchange format (RIF³⁹) has been established to enable distributed services for using rule-based knowledge from different sources. In general, a rule-based system uses a rule base to derive new facts from a given fact base or facts provided interactively by a user. However, surveys of the preferences of clinicians related to the use of such rule-based systems have identified the importance of the understandability of the reasoning used by these systems as an important factor for their acceptability [Teach and Shortliffe 1981].

The research of these decision support recommendation systems has evolved from focusing on standalone systems where clinicians are expected to enter all the required information about patients, to focusing on systems integrated into clinical information systems (CIS) so to easily and automatically retrieve information about patients, and now most recently to focusing on service-oriented systems that are expected to be able to connect to CIS without being integrated into these systems. This latest research model saves clinicians from the tedious work of entering information about patients that were already entered into CIS, while also maintaining the independence of the decision support system from CIS with the advantage being adaptability across a wide range and varieties of CIS [Wright and Sittig 2008].

³⁸ <http://neamh.cns.uni.edu/MedInfo/mycin.html>

³⁹ http://www.w3.org/2005/rules/wiki/RIF_Working_Group

Table 4.1: Notable Medical Diagnosis Expert Systems

Medical diagnosis System	Diagnosis Expert Area	Knowledge Representation	Reference
MYCIN	Antimicrobial selection for patients with bacteraemia or meningitis	Rule-Based (BC)	[Buchanan and Shortliffe 1984]
PIP	Renal disease	Frame-Based	[Pauker <i>et al.</i> 1976]
INTERNIST-1	Internal Medicine	Relational Database	[Miller <i>et al.</i> 1982]
ONCOCIN	Clinical Oncology	Rule-Based (BC)	[Shortliffe <i>et al.</i> 1981]
MDSS	Diabetes	Rule-Based (BC)	[Shortliffe 1987]
FuzzyDiagnose	Six Diseases	Fuzzy-Neural Net	[Moein <i>et al.</i> 2008]
MADHS	Traditional Chinese Medicine	Multiple Agents	[Qiao Yang Shieh 2008]

BC means backward chaining or goal-directed reasoning.

In this chapter, we will present our rule-based diagnostic system based on the notion of differential diagnosis (DDx) recommendation. Our system is composed of two cooperating disease diagnosis recommendation approaches. The first approach we call the ***Evidence-based DDx*** approach. Basically, rules are employed to represent clinical pathways⁴⁰, which are used by the health care providers to describe the processes of disease diagnosis and treatment. The idea is to design an adaptive rule-based DDx recommender that can adapt to changes in clinical pathway knowledge. The recommender systematically guides clinicians, according to the diseases symptoms ontology (DSO), to identify possible diagnoses, and prompts clinicians, according to clinical pathways rules, to provide lab test results in order to determine a diagnostic decision. The second approach is called the ***Proximity-based DDx*** approach. It utilizes semantic web technologies for navigating, with the aid of the DSO and a patient ontology (PO), clinical documents to extract important clinical and diagnostic variables. To find data for these variables, it uses diagnostic data produced by the first approach as well as from sound diagnostic datasets. Data mining techniques will then be applied to the data in order reach a diagnostic decision.

Both rule-based approaches employ challenging semantic web technologies which identify them as notable solutions for the problem of disease diagnosis. Both approaches build on our progress and methodologies developed in our previous chapters. Although clinicians may use either type to analyze a disease diagnosis case study, the evidence-based DDx recommendation approach must be the first analytic option in the absence of proper diagnostic datasets. In any case the evidence-based approach, when used frequently will generate more data that can be added to a dataset that will eventually grow to be used by the proximity-based approach. The evidence-based and proximity-based approaches for DDx recommendation can cooperate to form an

⁴⁰ http://en.wikipedia.org/wiki/Clinical_pathway

overall DDx recommendation model. Results generated by the evidence-based DDx recommender can be used for prediction by the data mining algorithms of the proximity-based DDx recommender. On the other hand, data mining algorithms from the proximity-based approach generate rules that can be used to update the rule base of the evidence-based approach. Figure 4.1 illustrates our vision of developing a DDx recommender based on the two approaches. Figure 4.2 illustrates the overall architecture of our rule based DDx recommender.

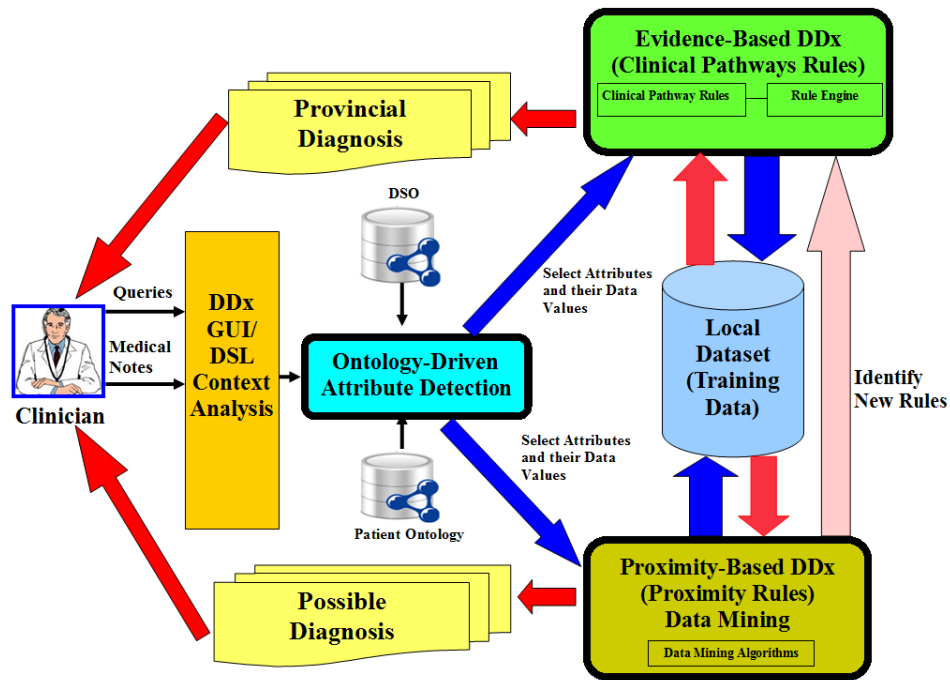


Fig. 4.1: The Integral Model of Evidence-based & Proximity-based DDx Recommendations

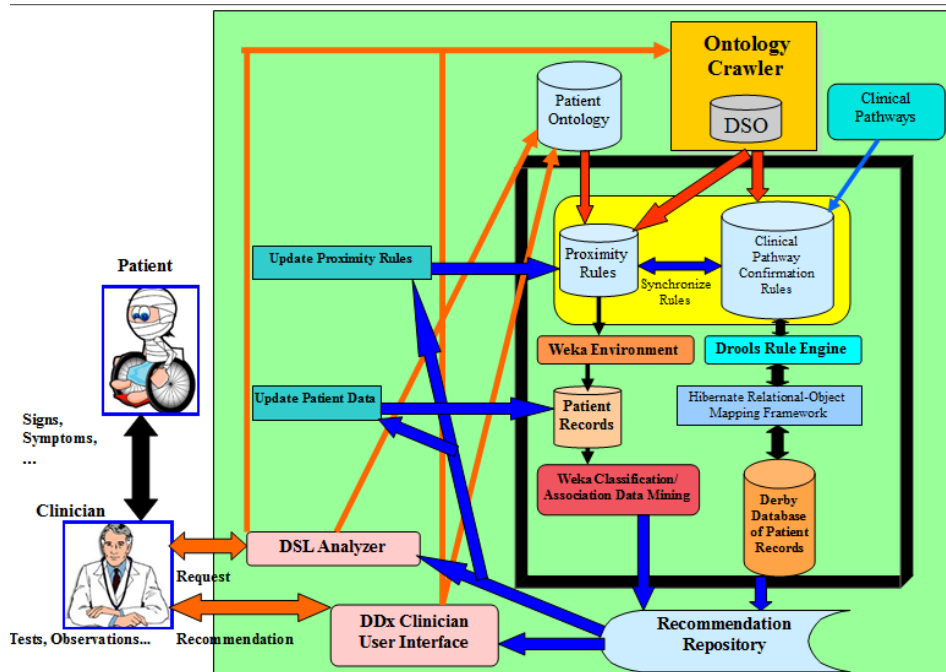


Fig. 4.2: Technologies of our DDx Rule-Based Recommender and their Connectivity

Clinical pathways are the blueprint for a plan of care representing the best clinical practice for specified groups of patients with a particular diagnosis. They are a documented sequence of clinical interventions that help a patient with a specific condition or diagnosis move progressively through a clinical experience to a desired outcome. Predominantly, they are management tools, clinical audit tools and recommenders that are based on sound clinical information which may change with time [Audimoolam *et al.* 2005]. Clinical Pathways are thought to improve the quality of patient care and make a more efficient usage of health resources [Fernandez-Llatas *et al.* 2010, Every 2000].

In this chapter, we describe a new method for representing and reasoning based on clinical pathways. Our representation enriches the event/evidence-based clinical logs with information about disease indicators using information provided from our DSO ontology crawler (Chapter 3). Using the evidence-based approach, it will be possible to start with the full collection of diseases, and zoom-in to a small enough subset of diseases for direct inspection. The reasoning used in our approach employs a forward chaining dynamic strategy to confirm a disease diagnosis based on the provincially identified diseases suggested by the DSO ontology crawler. The *clinical pathway rules*, of the evidence-based approach, determine a diagnosis through a series of interactions with the clinician to recommend and elicit medical tests, procedures and other information needed from the patient record for proving a diagnosis recommendation. The *proximity rules*, of our proximity-based approach, attempt to provide clinicians with approximate inferences about disease diagnosis since many of the signs and symptoms may involve fuzzy and incomplete concepts. In this direction, we rely on the power of knowledge discovery, available within the paradigm of data mining, from networks and graphs of symptoms, signs and lab tests.

Data mining knowledge discovery involves approximate predictions with variable degrees of certainty depending on the availability and quality of training data. Table 4.2 illustrates some of the standard algorithms used within the paradigm of data mining to perform diagnosis prediction (classification) as well as to provide new proximity diagnosis rules (association). Other attempts to define proximity rules include the use of fuzzy linguistic rules [Hasan *et al.* 2010], neural networks [Al-Shayea 2011] and predictive techniques based on data mining [Gorunescu 2007]. However, the diagnosis predictions based on these methods depend solely on static techniques that use the weights of the symptoms and rule order defined by the rule firing mechanism.

Table 4.2: A Sample of Notable Data Mining Classification and Association Algorithms

Algorithm	Java Version	Type	Description	Reference
C4.5	J48	Classification	Decision Tree-based algorithm	[Quinlan 1999]
IREP	JRIP	Classification	Rule-based algorithm	[Cohen 1995]
NaiveBayes	NaiveBayes	Classification	Probabilistic classification algorithm	[Domingos 1997]
NBTree	NBTree	Classification	A hybrid classification algorithm combining the Naive Bayes algorithm with any other classification algorithm which is decision-tree based	[Kohavi 1996]
Apriori	Apriori	Association	A classic algorithm for producing association rules	[Agrawal 1994]

4.2. Incorporating Clinical Pathways for Medical Diagnosis Recommendation

Clinical pathways are guidelines and recommendations for the treatment and management of diseases generated by a team of clinical professionals, with a clear clinical intention or goal – either to diagnose a disease, or to perform a treatment [Chen 2006]. Clinical pathways are designed for a group of patients with similar needs, and are very useful for differential diagnosis [Nikoskelainen 2005].

Clinical pathway knowledge is dynamic and requires continuous adaptation and customization in order to become easily usable by an adopting institution as a guideline for treating a group of patients. In this sense, clinical pathways require a reasoning engine that can cope with dynamic data [Alexandrou *et al.* 2009]. For this reason, traditional expert systems cannot easily be used for modeling clinical pathways as their knowledge representation (e.g. static rules) and reasoning (e.g. backward chaining) techniques rely on pre-built approaches and respond only to specified queries. One approach to cope with the dynamic nature of clinical pathways is to employ an innovative adaptive rule-based engine, which can handle the implantation of evolving clinical pathways knowledge into its rule base. By implantation we mean directly transferring clinical pathway knowledge into rules compatible with a particular adaptive rule-based system.

Another approach to integrating clinical pathways into medical decision support systems is to guide their dynamic variation through ontologies. There are attempts in their infant stage to generate an ontology-based approach for clinical pathways representation. Instead of directly translating clinical pathways into rules for a specific rule engine, ontologies can be constructed to represent clinical pathway knowledge in a standard and relational way. Then, the resulting clinical pathway ontology (CPO) is translated into rules that are compatible with a specific adaptive rule-based system. It is easy to convert ontology-based rules to knowledge representation specific to any rule engine. On the other hand, it is much more difficult to convert rules from one format into another. However, engineering, specifically designing and implementing, ontologies for clinical pathways is an important challenge [Chabalier *et al.* 2007, Ye *et al.* 2009]. The best notable ontology known to date is the KEGG Pathways⁴¹ database representing molecular dataset pathway maps used for biological interpretation of higher-level systemic function. Lin [Lin 2009] attempted to develop such flexible clinical pathways by adding meta rules for updating a clinical pathway ontology. Many other researchers followed the same ontological approach [Chen *et al.* 2004, Chen 2006, Popescu *et al.* 2004, Ye *et al.* 2009, Hu *et al.* 2011, and Huang *et al.* 2011]. Another variation of the ontological approach is combining CPO along with SWRL⁴² (Semantic Web Rules Language) rules to handle exceptional scenarios in clinical pathway execution [Alexandrou *et al.* 2009, Alexandrou *et al.* 2008]. However, these research attempts have not discussed in detail the structure of their proposed clinical pathway ontologies. They have not provided the implementation of clinical pathway ontologies. In the absence of publicly available implementations of clinical pathways ontologies, the implantation approach represents the only viable approach for implementing clinical pathways.

In this chapter, we are developing, as part of the evidence-based approach for disease diagnosis prediction, truly flexible clinical pathways using easily editable rules that can be updated frequently by the clinicians as well as the system developers. The rules we developed are from several clinical pathways for diabetes [Victoria Dept. of Health 2009], hypertension [NHS 2004], anemia [Goodnough *et al.* 2005], and calcemia. We call these rules the *clinical pathways confirmation rules*. The confirmation rules take their data from a relational patient database that stores and manages the dynamically changing clinical tests and data. The DSO ontology crawler assists in selecting the right group of confirmation rules. It does so by selecting the clinical pathway rules for diseases that show certain observed patient symptoms. Thus the confirmation rules are agile in two senses: the rule selection is directed by the DSO ontology, and the rules utilize data from a frequently changing clinical database. Figure 4.3 illustrates the evidence-based DDx recommendation model. We call a rule engine based on dynamic updateable clinical pathways rules, and supported by diagnosis ontologies for context-aware reasoning an evidence-based DDx recommendation model. The word "*evidence*" refers to the clinical pathway.

⁴¹ <http://www.genome.jp/kegg/pathway.html#global>

⁴² <http://www.w3.org/Submission/SWRL/>

Our DSO and PO ontologies contribute significantly to our evidence-based DDx recommendation model. As new clinical data arrives, the two ontologies select only data relevant to the diagnosis of a specific case. They then feed the selected data to the patient database, which in turn passes the data to the rule engine. Upon receiving the selected data, the rule engine would then compare it to the clinical pathway confirmation rules. This process considerably enhances the reasoning and data selection mechanisms. It optimizes the rule firing process by refining the data input to only include data relevant to a specific diagnostic case.

The intelligence of our clinical pathways and ontology driven evidence-based recommender is complemented by the interactivity with clinicians. First of all, interactivity with the physician is essential for any medical recommendation system. During the start of the differential diagnosis process, our ontology driven recommender guides the clinician to prepare a list of possible diagnoses. As pointed out in section 3.2, our DSO crawler provides clinicians with relational queries that guide them in preparing a list of possible diagnoses, and in choosing the right clinical pathways. When the ontology driven recommender proposes a list of possible diagnoses, the clinician should be able to order the system to rule out one or some of the possible diseases, include diseases not given by the recommender or even select one of the diseases as the diagnosis based on knowledge not available to the system. This iterative interaction, in the process of differential diagnosis, between the clinician and the recommender greatly improves the chances of accurate diagnosis.

As the system guides the clinician along the clinical pathways of a disease, it asks the clinician for information such as test results and based on that information decides to move closer towards a diagnosis by taking one of the pathways. The element of interactivity here is that the clinician is given the power to use his/her cognition to approve or disapprove each rule-based decision, taken by the recommender, to select one pathway or another in the clinical pathways. If the clinician approves a decision, the recommender moves forward along the pathway. If the clinician disapproves a decision, the physician selects an alternative path/decision and the recommender moves forward with the diagnosis process based on this decision. The clinician should be able to update, override, or define exceptions to the rules used by the recommender based on medical experience or interacting with the recommender. A key interaction is that each diagnosis made by the evidence-based recommender must be confirmed by the clinician. Actually, our evidence-based component of the DDx recommender will not be able to update the data files of the proximity-based recommender (described in section 5) before the authentication and confirmation of the clinician. The defining criteria for interactivity is to engage the physician to make use of his/her cognitive knowledge in the diagnosis process better yet to learn from his/her heuristic knowledge, learn from clinical data, and to verify medical diagnosis decisions. This interactive iterative process of differential diagnosis allows the recommender to become dynamic, context-sensitive, and clinician-centric as well as to yield adaptive and evolving clinical pathways. This added value process sets the DDx recommender apart from traditional static expert systems or clinical support systems. Actually adding the interactivity feature to the

clinical pathways is very important as many of the diseases pathways are increasingly complex and the clinicians are overwhelmed with information and have no time to spend researching them. Without the element of interactivity, clinicians may send a patient down the wrong path which can result in less than optimum clinical outcomes.

Figure D1, a sequence diagram, in **Appendix D** is another way to illustrate the evidence-based diagnosis recommendation model.

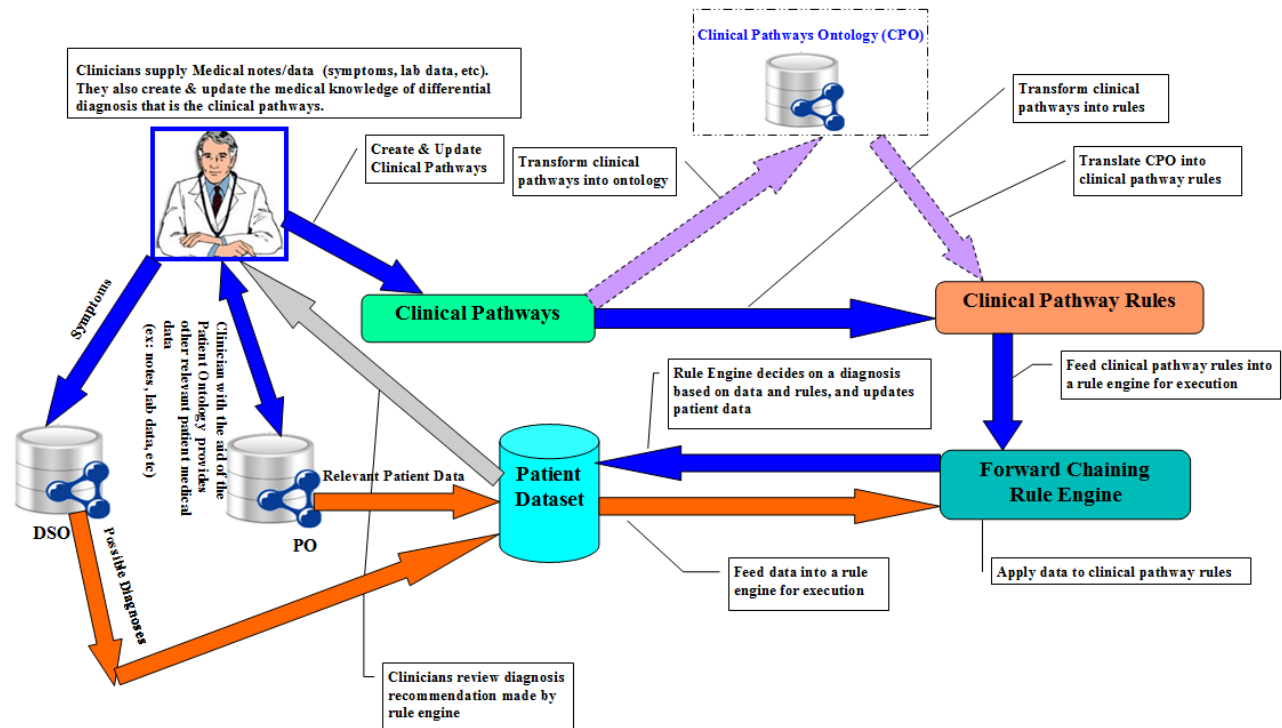


Fig 4.3: Illustration of Evidence-based Diagnosis Recommendation Model

4.3. Incorporating Proximity Rules for Medical Diagnosis Recommendation

Many techniques have been proposed to build disease diagnosis models using the notion of proximity. Table 4.3 lists some of the notable attempts:

Table 4.3: Proximity Models Used in Disease Diagnosis Recommendation

Model Name	Reference
Fuzzy Logic	[Adlassnig 1980]
K-Nearest Neighbour Neural Network	[Peterson 2009]
Naïve Bayes Classifiers	[Isam <i>et al.</i> 2007]
Rough Sets	[Anderson 2007]
Confidence Association Rules	[Gamberger <i>et al.</i> 1999]

In all these models except confidence association rules (data mining), complex mathematical derivations that cannot easily be employed for the process of differential diagnosis are needed. To support a medical expert in discovering clinically useful knowledge from ill-defined attributes and missing values as in most cases in disease diagnosis, one needs to avoid strict mathematical formulation requiring an exact number of parameters. With confidence association rules, however, proximity is introduced at a simplistic level using probabilistic thresholds attached to each knowledge rule or what is called confidence factors. These factors represent static indicators of the likelihood of a rule's applicability and success. Estimating these factors is a challenging research problem as there are no universal static factors that can fit all types of diagnostic problems [Balcázar 2009].

With data mining one can build a model according to the continuous change in the clinical process. However, the classical data mining techniques require proper understanding of the data in advance [Famili and Ouyang 2003]. Therefore, the data mining process will not be effective without the availability of a training dataset as well as clear knowledge of the data hierarchies. For the purpose of introducing knowledge about the data hierarchies, there are some recent research attempts to use ontologies in directing the prediction of the data mining techniques [Brisson and Collard 2008]. However, there is no ontology-driven data mining approach for disease diagnosis recommendation. For this purpose, this chapter introduces a new method for extracting proximity rules driven by a patient ontology that we developed from generic patient data records. The use of our patient ontology will help clinicians to extract relevant datasets from any available silo of patient records. Our patient ontology contains 241 major classes. These classes represent various patient attributes divided in four major categories: patient medical condition, patient allergies, patient medical history, and patient information (age, height, etc.). The medical condition category includes condition name, associated symptoms and lab tests. The patient medical history includes history of hospitalization, history medical conditions, family medical history, and any record of patient drinking or smoking. This is an OWL type where we published an online version of it at our university flash server (<http://flash.lakeheadu.ca/~omohamme/PatientOntology.owl>). Figure 4.4 provides a snippet of our patient ontology.

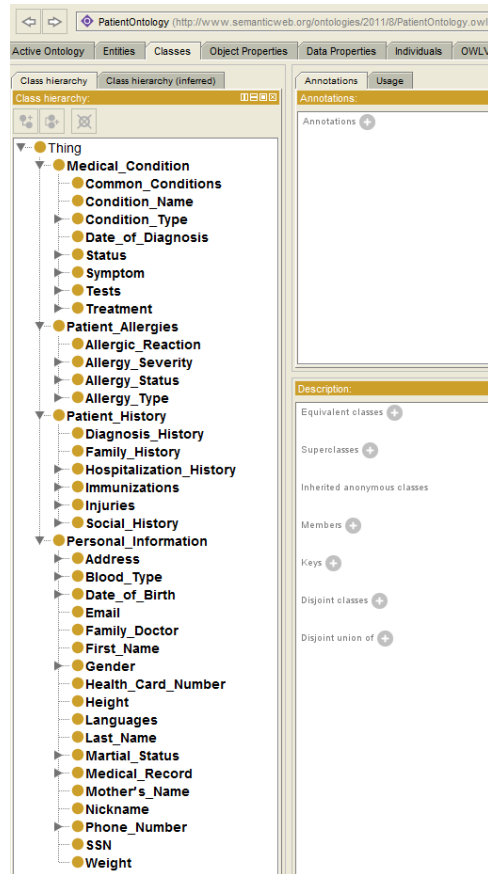


Fig. 4.4: Snippet of Patient Ontology as viewed by Protégé Ontology Editor

Clinicians need to extract, from patient data records, the relevant symptoms and test data for a certain diagnostic case. Using the proximity-based DDX recommendation approach, the patient and DSO ontologies will select relevant patient attributes, lab test attributes, and certain possible diseases under consideration for diagnosis. Training data for the selected clinical variables will be then used by data mining classification algorithms to learn diagnosis trends. After that these classification algorithms use the learned diagnosis trends to predict the diagnosis for provided test data (i.e. new clinical data cases that require diagnosis). Note that learned diagnosis trends are only valid predictors of diagnosis for test data, when the clinical attributes of the test data largely match the clinical attributes of the training data. The association algorithms use the training data to learn rules relating diagnosis variables (clinical attributes) and other diagnosis variables, and rules relating diagnosis variables and possible diagnoses. These newly generated rules can be used to make diagnosis predictions/recommendations. These newly generated rules can also be added to, or can override or update clinical pathway rules from the evidence-based approach. Using the proximity-based DDX recommendation model, clinicians can ask the following queries in order to conduct both association and classification data mining processes:

1. Find whether or not a patient has a certain disease based on the patient's demographic information, and lab test results. (Classification using Test Data)

- Find associations/rules describing the relationship between a patient's demographic information and lab test results on one hand, and a certain possible diagnosis on the other hand. (Association Rules)

Figure 4.5 illustrates our vision of employing an ontology-driven data mining (proximity) approach for providing DDX recommendations. The approach uses classification algorithms to predict diagnosis for new test cases based on some available training data. Also based on the available training data, it uses association algorithms to find rules for diagnosis recommendation for the diagnosis cases in the training data.

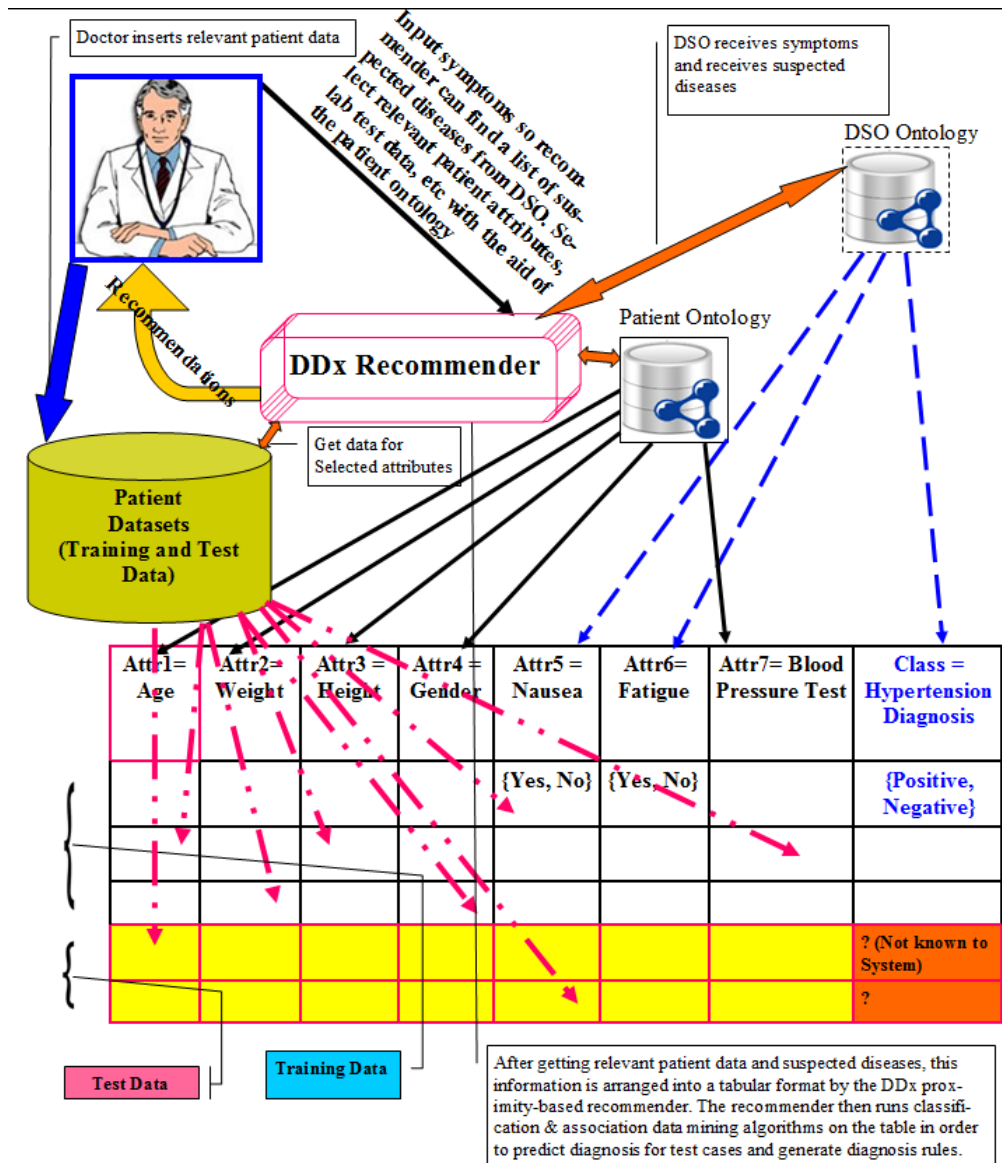


Fig. 4.5: Ontology-Driven Data Mining (ODDM) Model for DDX Recommendation

Figure D2, a sequence diagram, in **Appendix D** is another way to illustrate the proximity-based diagnosis recommendation model.

Clinicians use the ODDM DDx recommender to obtain diagnostic recommendations based on previously available training data. The ODDM DDx recommender attempts to construct a model for the training data (e.g. Decision Tree or Association Rules) and uses this model to determine the diagnosis recommendation for the new cases. Interaction with clinicians is crucial for this model. Interaction with the clinician in this model is done through data. Clinicians provide diagnosis data to the proximity-based DDx recommender, which uses to make diagnosis predictions and rules through its data mining algorithms. Once a prediction is made by the recommender, the prediction is shown to the clinician which decides to either accept or modify (or correct) the recommendation. In either case, the diagnosis accepted by the clinician will be added to the database used by the recommender. Every time a confirmed diagnosis case is added to the database, the knowledge base of the recommender widens allowing it to find more accurate trends among the diagnosis cases and to learn from the knowledge of the clinicians.

Our ODDM DDx recommender stores the test data in a format that is identical to the training dataset format. We chose the ARFF format which defines clearly the data and its description (i.e. metadata) in one file. The data mining algorithm is the mechanism that creates a data mining model. To create a model, an algorithm first analyzes a set of training data (dataset) and looks for specific patterns and trends. The algorithm uses the results of this analysis to define the parameters of the mining model. These parameters are then applied across the entire test data to extract actionable patterns and detailed statistics. The mining model that an algorithm creates can take various forms including⁴³:

- A set of rules (e.g. describe how products are grouped together in a transaction)
- A decision tree (e.g. predicts whether a particular customer will buy a product)
- A mathematical model (e.g. forecasts sales)
- A set of clusters that describe how the cases in a dataset are related

There are many available Java Data Mining APIs (e.g. Oracle JDM⁴⁴, JDMP⁴⁵, Weka⁴⁶) which we can incorporate into our DDx recommender. However, we chose the Weka API [Bouckaert *et al.* 2010] because it is open-source and their data mining classes can be incorporated in any Java implementation like our DDx recommender. Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine

⁴³ <http://www.globaldataconsulting.net/bi-stickers/microsoft-data-mining-algorithms>

⁴⁴ http://download.oracle.com/docs/cd/B28359_01/datamine.111/b28131/java_api.htm

⁴⁵ <http://www.jdmp.org/documentation/api-docs/>

⁴⁶ <http://www.cs.waikato.ac.nz/ml/weka/>

learning schemes⁴⁶. In **Appendix A**, we show how we used the Weka API for data mining work as part of our proximity-based DDX recommendation process.

4.4. Selecting a Rule Engine for Medical Diagnosis Recommendation

Many clinical applications have to deal with the dynamic changes of clinical pathways. A solution is to have a rule engine, which is basically a set of tools that enable clinicians and developers to build decision logic based on clinical data. The rule engine applies rules and actions as defined by end users without affecting how the application runs. The application is built to deal with the rules, which are designed separately. Examples of rule engines include Drools⁴⁷, Fair Isaac Blaze Advisor⁴⁸, ILOG JRules⁴⁹, and Jess⁵⁰, to name a few. The lack of standards, however, may be a major factor in deterring businesses from using rule-based applications. Most rule engines have proprietary APIs, making them difficult to integrate with applications. If a rule engine is no longer supported and the business decides to adopt another rule engine, most of the application code will need to be rewritten. However, the JSR 94 is an attempt to standardize rule engine implementations for Java technology. The four rule engines mentioned earlier support JSR 94. JSR 94 provides guidelines for the rule administration and rule runtime APIs, but it defines no guidelines for what language to use to define the rules and actions⁵¹.

The underlying idea of a rule engine is to externalize the business or application logic. A rule engine can be viewed as a sophisticated interpreter of if-then statements. The if-then statements are the rules. A rule engine is a great tool for efficient decision making because it can make decisions based on thousands of facts quickly, reliably, and repeatedly⁵².

Among the four notable JSR 94 compatible rule engines, we choose Drools. Drools is an open source rules engine, written in the Java language, which uses the Rete algorithm⁵³ to evaluate the rules you write. Drools rule engine lets you express your business logic rules in a declarative way. You can write rules using a non-XML native language that is quite easy to learn and understand. Also you can embed Java code directly in a rules file, which makes Drools rules even more expressive and flexible.

⁴⁷ <http://www.jboss.org/drools>

⁴⁸ <http://www.w3.org/2004/12/rules-ws/paper/55/>

⁴⁹ <http://logic.stanford.edu/POEM/externalpapers/iRules/WP-JRules50Strengths.pdf>

⁵⁰ <http://www.jessrules.com/>

⁵¹ <http://java.sun.com/developer/technicalArticles/J2SE/JavaRule.html>

⁵² <http://java.sun.com/developer/technicalArticles/J2SE/JavaRule.html>

⁵³ http://en.wikipedia.org/wiki/Rete_algorithm

4.5. The Drools Rule Engine

With many traditional programming-based rule engines, complex rules are difficult to automate, especially when a single change to a rule can impact hundreds of rules and processes. Drools enable us to accurately automate and change even the most complex rules with relative ease, and integrate seamlessly into the Java environment.

4.5.1 Using the Drools Rule Engine

As mentioned in the previous section, using a rule engine can significantly reduce the complexity of components that implement the business-rules logic. Actually an application that uses a rules engine to express rules using a declarative approach has an even a higher chance of being more maintainable and extensible than one that doesn't¹⁹. With Drools developers and business users alike are able to implement the complex business logic for their application in a declarative manner [Olivieri 2008]. Drools examine objects to find patterns and uses rules that describe these patterns to invoke certain actions⁵⁴.

Drools is a reasoning engine that includes a forward chaining rule engine with a declarative form of rule representation. For the purpose of this thesis, we use the JBoss Rules Workbench IDE as a tool to write and test the rules. The JBoss Rules workbench is delivered as an Eclipse plugin, which allows you to author and manage rules from within Eclipse.

4.5.2 Drools Rule Firing Mechanism

Since Drools is a forward chaining rule engine, it reacts to incoming lab test data by checking rules that might fire and creates other data or to signal diagnosis. Figure 4.6 illustrates the general concept behind Drools rule firing mechanism. The firing mechanism requires the creation of a database and interface that allows clinicians to enter medical test data results. The rule firing mechanism is responsible to check dynamically if there is any update to the database which may trigger rules to fire. In the following details below we illustrate how we created the database and the dashboard for inserting medical test results by the clinicians.

⁵⁴http://www.cisco.com/en/US/docs/net_mgmt/active_network_abstraction/3.5.1/administration/user/guide/ruleseng.pdf

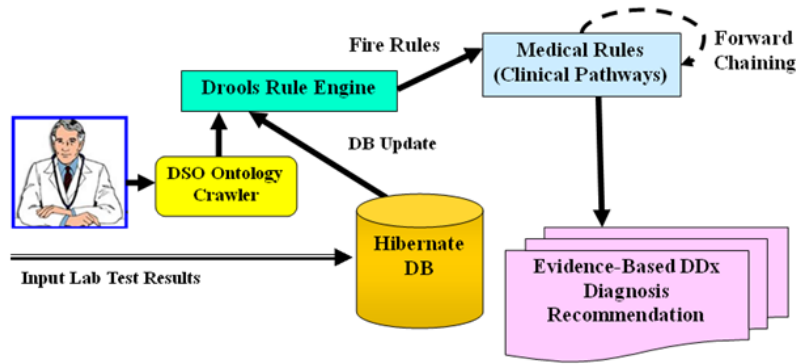


Fig. 4.6: Rule Firing Mechanism in the Evidence-Based Disease Diagnosis Recommender

The Database for Medical Lab Test Results

In our implementation we used the Apache Derby database⁵⁵ to create a modern database management system for storing medical test data results. However, the database uses its own server and thus will not affect the performance of our DDx recommender system.

The goal of our database is to link diseases to their corresponding lab tests. In other words for each disease define the lab tests needed to reach a diagnosis. First, we create a table (Table 4.4) of diseases chosen for our prototype. Second, we create a table (Table 4.5) of lab tests for the diseases defined in table 4.4. Finally, another table is needed for individual test result items for each lab test in table 4.6. Other tables are created to connect diseases to lab tests, and lab tests to test results.

Table 4.4: Diseases Table

Status	Result1	DIS_ID	DISEASE_NAME
1	1	1	Diabetes
2	2	2	Hypertension
3	3	3	Adult respiratory distress syndrome
4	4	4	Anemia
5	5	5	Calcemia

Table 4.5: Medical Lab Tests

Status	Result1	TEST_ID	TESTDESCRIPTION	MEDICAL_TEST_NAME	SEQUENCE_NUMBER
1	1	1	NULL	Fasting plasma glucose	1
2	2	2	The OGTT requires fasting for at least 8 hours before the test. The plasma glucose level is measured	Oral glucose tolerance	1
3	3	3	NULL	Random plasma glucose	1
4	4	4	NULL	Blood pressure	1
5	5	5	NULL	Arterial blood gas	1
6	6	6	NULL	Complete blood count	1
7	7	7	NULL	blood test	1

⁵⁵ <http://db.apache.org/derby/>

Table 4.6: Test Results

Status	Result1							
	TRID	AMOUNT	DESCRIPTION	DIAGNOSIS	NORMAL	RESULT	UNIT	TEST_JOINT
1	1	0.0	glucose - FPG	0	0	NULL	mg/dL	1
2	2	0.0	glucose - OGGT	0	0	NULL	mg/dL	2
3	3	0.0	glucose - Random plasma glucose	0	0	NULL	mg/dL	3
4	4	0.0	systolic	0	0	NULL	mmHg	4
5	5	0.0	diastolic	0	0	NULL	mmHg	4
6	6	0.0	ABG levels	0	0	NULL	PaO2	5
7	7	0.0	hemoglobin, female	0	0	NULL	mg/dL	6
8	8	0.0	hemoglobin, male	0	0	NULL	mg/dL	6
9	9	0.0	% RBC, male	0	0	NULL	mg/dL	6
10	10	0.0	% RBC, female	0	0	NULL	mg/dL	6
11	11	0.0	calcium levels	0	0	NULL	mg/dL	7

Inserting Test Results

When test results need to be added to the database, one needs to consult the database and insert the result at the right place. For example if we want to insert a test result into table 4.6 for the fasting plasma glucose test with the amount of 130 mg/dL, we need to consult the database and added to the table. The resulting updated table is shown in Table 4.7. When an update occurs to the database, the drools rule engine fires the medical rules, which in turn take the test results and make decisions regarding diagnosis recommendations. Actually, the rules engine is notified of an update to the database and will fire the matching rules. Table 4.8 illustrates the result of this update. As can be seen, the test results table is updated to include the diagnosis result of positive (Diagnosis = 1) in the test result entry of 130 mg/dL fasting plasma glucose (FPG). This means a test result of 130 mg/dL FPG indicates a positive diagnosis of diabetes.

Table 4.7: Updated Test Results Table

Status	Result1							
	TRID	AMOUNT	DESCRIPTION	DIAGNOSIS	NORMAL	RESULT	UNIT	TEST_JOINT
1	1	130.0	glucose - FPG	0	0	NULL	mg/dL	1
2	2	0.0	glucose - OGGT	0	0	NULL	mg/dL	2
3	3	0.0	glucose - Rando...	0	0	NULL	mg/dL	3
4	4	0.0	systolic	0	0	NULL	mmHg	4
5	5	0.0	diastolic	0	0	NULL	mmHg	4
6	6	0.0	ABG levels	0	0	NULL	PaO2	5
7	7	0.0	hemoglobin, fem...	0	0	NULL	mg/dL	6
8	8	0.0	hemoglobin, male	0	0	NULL	mg/dL	6
9	9	0.0	% RBC, male	0	0	NULL	mg/dL	6
10	10	0.0	% RBC, female	0	0	NULL	mg/dL	6
11	11	0.0	calcium levels	0	0	NULL	mg/dL	7

Table 4.8: Updated Test Results Table

Status	Result1							
	TRID	AMOUNT	DESCRIPTION	DIAGNOSIS	NORMAL	RESULT	UNIT	TEST_JOINT
1	1	130.0	glucose - FPG	1	0	positive test for diabetes	mg/dL	1
2	2	0.0	glucose - OGGT	0	0	NULL	mg/dL	2
3	3	0.0	glucose - Rando...	0	0	NULL	mg/dL	3
4	4	0.0	systolic	0	0	NULL	mmHg	4
5	5	0.0	diastolic	0	0	NULL	mmHg	4
6	6	0.0	ABG levels	0	0	NULL	PaO2	5
7	7	0.0	hemoglobin, fem...	0	0	NULL	mg/dL	6
8	8	0.0	hemoglobin, male	0	0	NULL	mg/dL	6
9	9	0.0	% RBC, male	0	0	NULL	mg/dL	6
10	10	0.0	% RBC, female	0	0	NULL	mg/dL	6
11	11	0.0	calcium levels	0	0	NULL	mg/dL	7

In the case of this example, the Drools rule below (Figure 4.7) will be fired. The rule will indicate a positive diagnosis of diabetes.

```
rule "3 - When a fasting plasma glucose test is performed to diagnose
Diabetes, then a result higher than 125 mg/dL or higher indicates positive
diabetes"
```

when

```
    t:Test(testName == "Fasting plasma glucose")
    d:Disease(diseaseName == "diabetes mellitus" || diseaseName ==
"diabetes mellitus type 2"
        || diseaseName == "diabetes mellitus type 1" || diseaseName ==
"diabetes" || diseaseName == "Diabetes")

    tr:TestResult(id == 1 && unit == "mg/dL" && description == "glucose -
FPG" && amount > 125)
```

then

```
    tr.setNormal(false);
    tr.setDiagnosis(true);
    tr.setResult("positive test for diabetes");
```

end

Fig. 4.7: Drools rule Fired in this Example

The two pieces of code below fire a state full knowledge session. In a state full knowledge session, all of the medical rules are examined and those rules which conditions are satisfied are fired in a forward chaining cycle. Figure 4.8 shows the main program, which fetches all the diseases, tests, and test results from the database tables above (Tables 4.4-4.8). Then these objects are passed to the rule firing function (line 29, figure 4.8). In figure 4.9, in line 21 the rules for the rule firing session are fetched from the drools rule file "medrules.drl" (see **Appendix A** for this rules file). In lines 28 & 29, these rules are fed into a state full knowledge session for rule firing. In a state full knowledge session for rule firing, when the condition for one rule is met and the rule fires taking some action or making some conclusion in the process, that action or conclusion can be used to satisfy the conditions for other rules. This process continues iteratively until there are no more rules to fire. This is basically the forward chaining process^{56 57}. After the rule firing session ends, the main program updates the database records (line 35, figure 4.8) to persist the conclusions made by the rules fired. The medical rules react with the disease, test, and test result objects to make conclusions about diagnosis.

⁵⁶ http://en.wikipedia.org/wiki/Forward_chaining

⁵⁷ [http://msdn.microsoft.com/en-us/library/aa349441\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/aa349441(v=vs.90).aspx)

```

package medRules.test;

import java.util.LinkedList;

public class Main {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        List<TestResult> results;
        List<Disease> diseases;
        List<Test> tests;

        RetrieveTestData data = new RetrieveTestData();

        diseases = data.getDiseases();
        results = data.getTestResults();
        tests = data.getTests();

        DiseaseTestRules dtRules = new DiseaseTestRules();

        dtRules.fireDiseaseTestRules(diseases, tests, results);

        UpdateTestResultData update = new UpdateTestResultData();

        for(int i = 0; i < results.size(); i++) {

            update.updateTestResultRecord(results.get(i));

        }

    }
}

```

Fig. 4.8: Main Program Firing Medical Rules

```

package medRules.test;

import java.util.LinkedList;

public class Main {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        List<TestResult> results;
        List<Disease> diseases;
        List<Test> tests;

        RetrieveTestData data = new RetrieveTestData();

        diseases = data.getDiseases();
        results = data.getTestResults();
        tests = data.getTests();

        DiseaseTestRules dtRules = new DiseaseTestRules();

        dtRules.fireDiseaseTestRules(diseases, tests, results);

        UpdateTestResultData update = new UpdateTestResultData();

        for(int i = 0; i < results.size(); i++) {

            update.updateTestResultRecord(results.get(i));
        }
    }
}

```

Fig. 4.9: Drools Rule Engine - State full Knowledge Rule Firing Session

4.6. Medical Data & Data Pre-processing for Proximity-Based Diagnosis Recommendation

4.6.1 Selecting Suitable Medical Dataset for Proximity-Based Diagnosis Recommendation

Data Mining approaches are used to extract meaningful information from data stored in large databases. Medical databases/datasets has been used so far by data mining processes in the area of disease diagnosis only for disease prediction as such a task becomes important in a variety of applications such as health insurance, tailored health communication and public health. Disease prediction is usually performed using publically available datasets such as HCUP⁵⁸, NHANES⁵⁹ or MDS⁶⁰ that were initially designed for health reporting or health cost evaluation but not for

⁵⁸ <http://www.hcup-us.ahrq.gov/>

⁵⁹ <http://www.cdc.gov/nchs/nhanes.htm>

⁶⁰ http://www.resdac.org/mds/data_available.asp

disease prediction [Popescu and Khalilia, 2011]. Healthcare Cost and Utilization Project (HCUP, pronounced "H-Cup") is a family of health care databases and related software tools and products developed through a Federal-State-Industry partnership and sponsored by the Agency for Healthcare Research and Quality (AHRQ). HCUP databases bring together the data collection efforts of State data organizations, hospital associations, private data organizations, and the federal government to create a national information resource of patient-level health care data. HCUP includes the largest collection of longitudinal hospital care data in the United States, with all-payer, encounter-level information beginning in 1988. These databases enable research on a broad range of health policy issues, including cost and quality of health services, medical practice patterns, access to health care programs, and outcomes of treatments at the national, State, and local market levels⁶¹. The National Health and Nutrition Examination Survey (NHANES) dataset is a collection of studies designed to assess the health and nutritional status of adults and children in the United States²⁶. The Long Term Care Minimum Data Set (MDS) is a standardized, primary screening and assessment dataset of health status which forms the foundation of the comprehensive assessment for all residents of long-term care facilities certified to participate in Medicare or Medicaid. The MDS contains items that measure physical, psychological and psycho-social functioning.

The HCUP NIS dataset represents data of hospital inpatient stays, containing data from about a thousand hospitals that constitute a 20% stratified sample of US community hospitals. There is one dataset for each year from 1988 to 2009 each containing from five to eight million stays. We will mainly be using data from the NIS core database for our data mining work within our proximity-based DDX recommendation model. We obtained the license for the NIS 2009 data from the HCUP Central Distributor under a data use agreement, and a licence number for use of the data given to us after taking for a 15-min mandatory online course on proper use of the data.

For each NIS data record (i.e. hospital visit), there are 126 clinical and nonclinical data elements⁶². Nonclinical elements include patient demographics, hospital identification, admission date, zip code, calendar year, total charges and length of stay. Clinical elements include procedures, procedure categories, diagnosis codes and diagnosis categories. Every record contains a vector of 15 diagnosis codes (1 primary diagnosis and 14 secondary diagnoses). The diagnosis codes are represented using the International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM)⁶³. The International Statistical Classification of Disease is designed and published by the World Health Organization (WHO). The ICD-9 codes are alphanumeric codes, 3-5 characters long and used by hospitals, insurance companies and other facilities to describe health conditions of the patient. Every code represents a disease,

⁶¹ <http://www.hcup-us.ahrq.gov/overview.jsp>

⁶² http://www.hcup-us.ahrq.gov/db/nation/nis/NIS_Introduction_2006.jsp

⁶³ <http://www.cdc.gov/nchs/icd/icd9cm.htm>

condition, symptom, or cause of death. There are numerous codes, over 14,000 ICD-9 codes and 3,900 procedures codes.

The NIS 2009 archive, containing data only from year 2009, is divided into four distinct databases [Concaro *et al.*]:

- (1) The *Inpatient Core database*, containing the inpatient discharge-level data; no patient identifier is provided in such data set, so that no patient level analysis is allowed;
- (2) The *Hospital Weights database*, recording data and attributes concerning the sampled hospitals;
- (3) The *Disease Severity Measures database*, containing information from different sets of disease severity measures concerning the inpatient stays;
- (4) The *Diagnosis and Procedure Groups database*, containing data elements and attributes concerning diagnosis and procedures included in the Inpatient Core records.

4.6.2 Data Pre-processing, Transformation and Filtering

In order to assist our proximity-based DDx recommendation, two stages of data pre-processing need to be done. The first stage of the data pre-processing we call Java pre-processing and is done using Java programs. The second stage we call Weka pre-processing and is done using the Weka data pre-processing tab. The first stage involves arranging the data into an ARFF file that can be read by Weka. With the NIS data, this involves several steps. First, the data is first converted into SAS format (table format) from its native ASCII format. Then, using a Java API called SassyReader⁶⁴ we retrieve data from the SAS table format. Then using Java's file writing classes and methods, we arrange the data into ARFF format. We select certain attributes we are interested in from the NIS data. Then, we create a header where each selected data variable is determined as attribute or class, given a name, and a type. Then the data is arranged into (patient) records with each variable value separated by a comma. The diagnosis data is encoded using ICD-9-CM. To decode the diagnosis information, a list of ICD-9-CM diagnosis codes and their corresponding diagnosis is needed. This list is provided at the NIS website⁶⁵. A Java program uses this list to decode the diagnosis information and store it into the ARFF data file. Some string manipulation is also required. This is done in several stages. The following diagram (Figure 4.10) clarifies this process.

⁶⁴ <http://sassyreader.eobjects.org/>

⁶⁵ <http://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp#download>

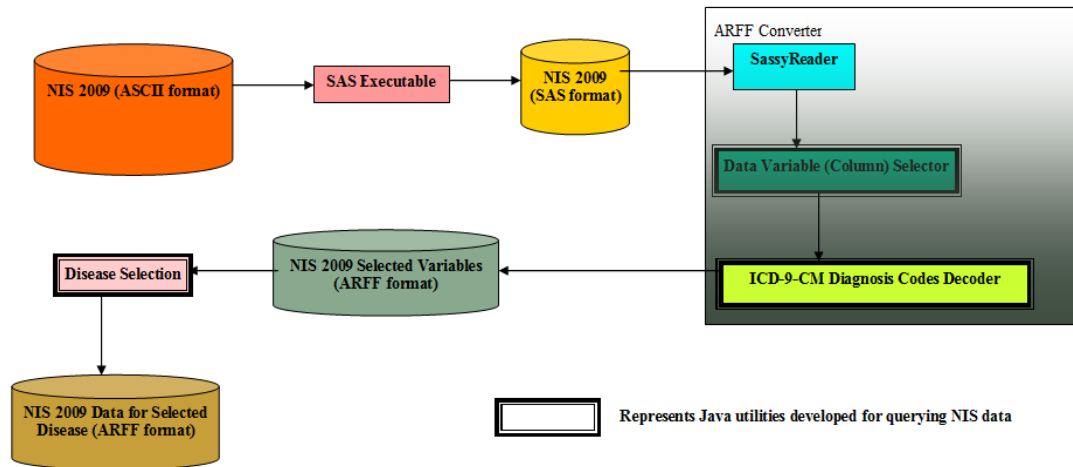


Fig. 4.10: NIS 2009 Data Format Conversion, Decoding, and Querying (Pre-processing)

The last data preparation that we need to perform involves filtering relevant data models for diagnosis. Since we are concentrating our efforts on certain types of diseases, we developed five filters to extract relevant data from NIS. Tables 4.9 to Table 4.13 provide the filtered format for four main diseases (2 Tables for Type 2 Diabetes, 1 for Anemia, 1 for Hypertension and 1 for Calcemia).

Table 4.9: Diabetes Data 1 (Number of records = 63)

Variable Name	Variable Type (Attribute or Class)	Possible Values for Variable	Description
ageYears	Attribute	0-125 years	Patient's age in years
ageDays	Attribute	0-365 days	Patient's age in days if patient is less than a year old
Gender	Attribute	0.0 (Male), 1.0 (Female)	Patient's gender
fasting_plasma_glucose	Attribute	All real numbers in the range 60-150 mg/dL	Glucose levels in the patient's blood
prim_diag	Class	Normal, Diabetes Type 2	Whether the patient has type 2 diabetes or not

Table 4.10: Diabetes Data 2 (Number of records = 768)

Variable Name	Variable Type (Attribute or Class)	Possible Values for Variable	Description
Preg	Attribute	0-17 times	Number of times pregnant
Plas	Attribute	0-199 mg/dL	Plasma glucose concentration in an oral glucose tolerance test
Pres	Attribute	0-122 mmHg	Diastolic blood pressure of the patient
Skin	Attribute	0-99 mm	Triceps skin fold thickness
Insu	Attribute	0-846 mu U/ml	2-Hour serum insulin
Mass	Attribute	All real numbers between 0-67.1	Body mass index (BMI)

Pedi	Attribute	All real numbers between 0.078-2.42	Diabetes pedigree function
Age	Attribute	21-81 years	Patient's age in years
Diagnosis	Class	0 (Normal), 1 (Diabetes)	Whether a patient has diabetes or not

Table 4.11: Anemia Data (Number of records = 117)

Variable Name	Variable Type (Attribute or Class)	Possible Values for Variable	Description
ageYears	Attribute	0-125 years	Patient's age in years
ageDays	Attribute	0-365 days	Patient's age in days if patient is less than a year old
Gender	Attribute	0.0 (Male), 1.0 (Female)	Patient's gender
Rbc	Attribute	All real numbers in the range 25-49 %	Percentage of red blood cells in the blood
Haemoglobin	Attribute	All real numbers in the range 7.7-17.2 mg/dL	Amount of hemoglobin in the blood
prim_diag	Class	Normal, Anemia	Whether a patient has anaemia or not

Table 4.12: Blood Pressure Data (Number of records = 834)

Variable Name	Variable Type (Attribute or Class)	Possible Values for Variable	Description
ageYears	Attribute	0-125 years	Patient's age in years
ageDays	Attribute	0-365 days	Patient's age in days if patient is less than a year old
Gender	Attribute	0.0 (Male), 1.0 (Female)	Patient's gender
systolicBloodPressure	Attribute	All integers between 60-169 mmHg	Systolic blood pressure of a patient
diastolicBloodPressure	Attribute	All integers between 40-119 mmHg	Diastolic blood pressure of a patient
prim_diag	Class	Hypotension, Normal, Hypertension	Whether a patient's blood pressure is low (Hypotension), normal, or high (Hypertension)

Table 4.13: Calcemia Data (Number of records = 21)

Variable Name	Variable Type (Attribute or Class)	Possible Values for Variable	Description
ageYears	Attribute	0-125 years	Patient's age in years
ageDays	Attribute	0-365 days	Patient's age in days if patient is less than a year old
Gender	Attribute	0.0 (Male), 1.0 (Female)	Patient's gender
calcium_levels	Attribute	All real numbers between	Calcium level in a patient's

		7.8-13.4 mg/dL	bloodstream
prim_diag	Class	Hypocalcemia, Normal, Hypercalcemia	Whether the patient has low calcium levels in the bloodstream (Hypocalcemia), or normal levels, or elevated levels (Hypercalcemia)

The second stage basically involves using Weka filters. Specifically, we use the **discretize** filter to transform continuous data variables in discrete variables. This is needed for the association algorithms to be able to analyze data.

4.7. Weka Data Mining Tool for Proximity-Based Diagnosis Recommendation

The Weka environment⁶⁶ (Bouckaert 2010), developed by the University of Waikato in New Zealand⁶⁷, is a collection of machine learning algorithms for data mining tasks. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. The Weka environment consists of two parts: a GUI interface and a Java API. The Java API is the engine Weka runs on as all the machine learning algorithms used by Weka are implemented in Java. The programmer can use the Java API to create simple interfaces to allow novice users, with little knowledge of data mining, to conduct data mining operations on their data. The results should reveal interesting trends in the data displayed in a way the novice user can understand. On the other hand, the GUI interface allows users to visualize and conduct filtering operations on data, apply different classification, regression, clustering, and association rules algorithms, and configure algorithm parameters. This allows the advanced user to conduct several experiments on a dataset to extract various trends. In this chapter, we will be using the GUI interface to show several data mining operations for proximity-based diagnosis recommendation. In this next chapter, we will be showcasing the Weka Java API as part of the prototype.

4.8. Testing the Proximity-Based Diagnosis Recommender

Since the Weka environment is easy to learn, clinicians may use the Weka GUI tool to conduct various pre-processing, classification, regression, clustering, association rules, and visualization operations on the medical data prepared in ARFF format. **Appendix A** illustrates the usage of Weka GUI toolkit for classification and association. In the next chapter (Chapter 5) instead of using Weka GUI, we are going to use an alternative approach where our prototype will take advantage of the Weka Java API. **Appendix B** lists diagnosis predictions for data records in tables 4.10-4.13 obtained using various classification algorithms provided by Weka. **Appendix**

⁶⁶ <http://www.cs.waikato.ac.nz/ml/weka/>

⁶⁷ www.waikato.ac.nz/

C lists diagnosis rules for data records in tables 4.10-4.13 obtained using the Apriori association algorithm provided by Weka. Both the Weka GUI and Weka API were used to obtain the results in **Appendices B & C**.

4.9. Validation of Evidence & Proximity-based DDx Recommendations

Our overall DDx recommendation model allows clinicians to use both the evidence-based and proximity-based DDx recommenders for assistance in diagnosing a specific patient's case. In rare cases, both recommenders for a specific patient's case may provide different diagnosis results. This is due to the fundamental difference between the two recommenders. As explained earlier, the evidence-based DDx recommender reaches a diagnosis decision based on applying clinical pathways rules to the data of the patient case under investigation. On the other hand, the proximity-based DDx recommender relies on data mining algorithms to extract rules from data of previous cases. These proximity rules are then applied to the data of the patient case under investigation in order to obtain a diagnosis. Consequently, the diagnosis results of the two recommenders may be different in some cases especially if the datasets used by the data mining algorithms of the proximity-based DDx recommender are not large and representative. If the datasets are representative, we expect that both recommenders' diagnosis results will most likely correlate.

In cases of differences in the predictions of the two recommenders, it is up to the clinician to examine both results and make his/her own decision about the patient's case. The clinician may choose one of the recommender's results or decide on a third result. The result indicated by the clinician is saved to the patient database. Both recommenders benefit from this process. A new confirmed diagnosis is added to the proximity-based DDx recommender's training dataset, which will help improve its accuracy of prediction. The new confirmed case would also update the set of proximity (association) rules of the specific disease that was under investigation. In the overall DDx recommendation model, the updated proximity rules are compared to the clinical pathway rules of the evidence-based recommender. In the case there are any differences between the two rule sets, the clinician would be allowed to compare these differences in the two rule sets. For each difference, the clinician can decide to support the proximity-based rules or the clinical pathway rules, or the clinician can also decide to support neither. If the clinician supports neither, he/she can write a new rule to settle the difference. The overall DDx recommender would replace the rules in both rule sets with the new rule written by the clinician. This is the spirit of our overall DDx recommendation model which is to give the clinician the power to make the final diagnosis decisions.

4.9.1 Evidence-based vs. Proximity-based Rules Comparison

In this section, we will examine a number of rule pairings. These rule pairings represent equivalent evidence-based and proximity-based rules. On the one hand, there is the evidence-

based rule(s) that describe a certain condition, and on the other hand there is proximity-based rule(s) that describe the same condition. By describe the same condition we mean that they refer to the same variable(s)/attribute(s) in their condition sections. So we have a pair of rules, and we call them equivalent because they describe the same condition.

We will look at the following two listings:

Both the evidence-based rule (listing 4.2) and the proximity-based rules (listing 4.3) refer to the same attribute (fasting plasma glucose) in their conditions sections. Therefore, these two rules are said to be equivalent but not exactly equal. The evidence-based rule says that if the fasting plasma glucose is less than 109.8 mg/dL then this should be considered normal. On the other hand, the proximity rules 7 & 17 (see listing 4.3) have the combined effect of indicating that if the fasting plasma glucose is less than 76.6 mg/dL then this is considered a normal result. Of course here the evidence-based result is the most accurate since it is based on prior medical knowledge (clinical pathways) while the proximity-based result is based on only 64 patient cases. It seems that significantly more patient cases are required to achieve a more accurate result. We say this because if we compare the proximity-based rules with the evidence-based rule we find a 30% error rate with the maximum normal FPG used by the proximity-based rules. If we look at the equivalent rules that check for the abnormal/diabetic FPG we find that the proximity-based rule (see listing 4.5), based on the same 64 patient cases, is closer to the evidence-based rule (see listing 4.4). The evidence-based rule states that if the FPG test result is higher than 125 mg/dL then the result indicates diabetes, while the proximity-based result states that an FPG result higher than 139 mg/dL indicates diabetes. The proximity-based rule here has an error rate of 17%.

However, we have found more encouraging results in our investigation of proximity-based association rules. We used a table of more numerous hypertension patient cases (840 patient cases) where patients underwent blood pressure testing and the results were either classified as normal, hypertensive (high blood pressure), or hypotensive (low blood pressure). The proximity-based rules produced by the association algorithm from the hypertension patient cases are more in line with the evidence-based rules based on the clinical pathways of hypertension. For example, the evidence-based rule in listing 4.6 states that a diastolic blood pressure of more than 90 mmHg is considered hypertensive. The corresponding equivalent proximity rules in listing 4.7 state that a diastolic blood pressure of more than 87.4 mmHg is considered hypertensive. The difference in the maximum normal diastolic blood pressure is only 3%. The effect is that these rules validate each other, and the proximity-based process is successful at producing accurate rules.

A full table comparing all the equivalent evidence-based and proximity-based rule sets is presented in **Appendix E. Table E1** provides a comparison of equivalent proximity and

evidence-based rule sets. The accuracy of the proximity-based rules as a function of the number of known patient cases available to the association algorithm is provided in **Table E2** and graphed in **Figure E1**.

```
rule "1 - If a fasting plasma glucose test is performed to diagnose Diabetes,
then a result 109.8 mg/dL or less indicates negative diagnosis or normal
glucose levels"
```

```
when
```

```
  t:Test(testName == "Fasting plasma glucose")
  d:Disease(diseaseName == "diabetes mellitus" || diseaseName ==
"diabetes mellitus type 2"
           || diseaseName == "diabetes mellitus type 1" || diseaseName ==
"diabetes" || diseaseName == "Diabetes")
```

```
  tr:TestResult(id == 1 && unit == "mg/dL" && description == "glucose -
FPG" && amount <= 109.8 && amount > 0)
```

```
then
```

```
  tr.setNormal(true);
  tr.setDiagnosis (false);
  tr.setResult ("Normal healthy glucose levels, no evidence of
diabetes");
```

```
end
```

Listing 4.2: Clinical Pathways Evidence-based Rule for Diabetes

```
7. fasting_plasma_glucose='(-inf-66.145453]' ==> prim_diagnosis=Normal  conf:(1)
```

```
17. fasting_plasma_glucose='(66.145453-76.643973]' ==> prim_diagnosis=Normal  conf:(1)
```

Listing 4.3: Proximity-based Rule for Diabetes

```
rule "3 - When a fasting plasma glucose test is performed to diagnose
Diabetes, then a result higher than 125 mg/dL or higher indicates positive
diabetes"
```

```
when
```

```
  t:Test(testName == "Fasting plasma glucose")
  d:Disease(diseaseName == "diabetes mellitus" || diseaseName ==
"diabetes mellitus type 2"
           || diseaseName == "diabetes mellitus type 1" || diseaseName ==
"diabetes" || diseaseName == "Diabetes")
```

```
  tr:TestResult(id == 1 && unit == "mg/dL" && description == "glucose -
FPG" && amount > 125)
```

```
then
```

```
  tr.setNormal(false);
  tr.setDiagnosis(true);
  tr.setResult("positive test for diabetes");
```

```
end Listing 4.4: A Second Clinical Pathways Evidence-based Rule for Diabetes
```

12. `fasting_plasma_glucose='(150.13361-inf)' 12 ==> prim_diagnosis=DIABETES UNCOMPL TYPE II conf:\(1\)`
 24. `fasting_plasma_glucose='(139.635091-150.13361]' ==> prim_diagnosis=DIABETES UNCOMPL TYPE II conf:\(1\)`

Listing 4.5: A Second Proximity-based Rule for Diabetes

```
rule "8 - If a blood pressure test is performed to diagnose hypertension,
then a result of more than 90 mmHg diastolic blood pressure would signal
hypertension"
```

when

```
  d:Disease(diseaseName == "Hypertension" || diseaseName ==
"hypertension")
  t:Test(testName == "Blood pressure" || testName == "blood pressure")
  tr:TestResult(id == 5 && unit == "mmHg" && amount >= 90 &&
(description == "Diastolic" || description == "diastolic"))
```

then

```
  tr.setNormal(false);
  tr.setDiagnosis(true);
  tr.setResult("hypertension - high diastolic blood pressure");
```

end

Listing 4.6: A Clinical Pathways Evidence-based Rule for Hypertension

62. `ageDays='All' DiastolicBloodPressure='(103.2-111.1]' ==> prim_diagnosis=HYPERTENSION NOS conf:\(1\)`
 67. `ageDays='All' DiastolicBloodPressure='(111.1-inf)' ==> prim_diagnosis=HYPERTENSION NOS conf:\(1\)`
 71. `DiastolicBloodPressure='(87.4-95.3]' ==> prim_diagnosis=HYPERTENSION NOS conf:\(1\)`
 89. `ageDays='All' DiastolicBloodPressure='(95.3-103.2]' ==> prim_diagnosis=HYPERTENSION NOS conf:\(1\)`

Listing 4.7: A Proximity-based Rule for Hypertension

4.9.2 Evidence-based vs Proximity-based Predictions

In this section, we compare prediction results for specific patient cases obtained from our evidence-based DDX recommender and from our proximity-based DDX recommender. **Appendix E** provides a comparison of predictions from both of the recommenders for two diseases (Anemia in **Table E3** and Diabetes in **Table E4**). The comparison indicates clearly that the predictions from both recommenders for most patient cases match. When comparing the accuracy of the proximity-based rules for anemia and calcemia in tables E3 & E4 with the accuracy of these rules in table E1, the accuracy is lower in tables E3 & E4. This simply means that the test cases in tables E3 & E4 need to be more carefully chosen to become more representative of the full range of values covered by the rules. We did our best to select representative test cases and they largely reaffirmed the high degree of accuracy of the proximity-based rules.

This chapter provided two approaches for differential diagnosis recommendation based on building an upper inferential engine on top of our Ontology Crawler described in chapter 3. The first approach is based on an evidence-based diagnosis strategy where disease diagnosis knowledge is stored in the form of rules and based on these rules one can identify any given diagnosis case. Drools rule engine and a database are used to construct the framework for the evidence-based part. Chapter 5 will explore the idea of using the framework based on our developed prototype for differential diagnosis where clinicians may interact with GUI to infer the diagnosis recommendation for any given case study. The second approach is the proximity-based approach where we employed data mining techniques for predicting diagnosis recommendation based on a large dataset. For this purpose, we have used the NIS licensed dataset and the Weka data mining environment for drawing diagnostic recommendations for a set of diseases that we used in chapter 3 (Diabetes, Anemia, Hypertension and Calcemia). We used several data mining algorithms to classify data as well as an association algorithm to infer the most interesting rules associating clinical variables and diagnosis. Similarly we are going to demonstrate in Chapter 5 how to use our developed prototype for proximity-based diagnosis recommendation. We trust that using the two approaches, clinicians can provide better diagnostic recommendations with more reliable accuracy.

Chapter 5: System Demonstration

5.1. Clinical Diagnosis Support Systems

Clinical diagnosis support software systems (CDSS) help clinicians in diagnosing clinical cases. Most of the clinicians are still relying on a manual clinical diagnosis process. A manual clinical diagnosis is a very complex, cumbersome and error prone process; even very experienced doctors sometimes fail to diagnose a clinical condition correctly at an early stage [Reddy 2010]. For this purpose, CDSS are considered important and useful to assist physicians and other health professionals with decision making tasks, such as determining diagnosis from patient data. However, modern CDSS need to be more interactive to help determine diagnosis, analysis, etc. of patient data. Traditional CDSS would literally make decisions for the clinician. The clinician would input the information and wait for the CDSS to output the "right" choice and the clinician would simply be expected to act on that output⁶⁸. The new methodology of using CDSS to assist forces the clinician to interact with the CDSS utilizing both the clinician's knowledge and the CDSS to make a better analysis of patients data than either human or CDSS could make on their own [Berner 2007]. The objective of this thesis is to use the notion of Differential Diagnosis (DDx) along with several emerging technologies from the paradigm of semantic web to develop a semantically enriched interactive CDSS model. We call this model the DDx recommendation model. The differential diagnosis methodology provides a systematic method for interactivity used to identify unknowns. This method, essentially a process of elimination, is used by physicians, nurse practitioners, physician assistants, and other trained medical professionals to diagnose the specific disease in a patient. In our model, the process of elimination is guided by the DSO ontology along with the clinical pathways rules. However, not all medical diagnoses follow strict pathways with available lab tests as some diagnoses are based on intuition or estimations of likelihood. For this purpose, our work considers another type of differential diagnosis that uses proximity in decision making. Our proximity-based DDx recommender uses a focused data mining approach for providing possible diagnoses. We call the focused approach as Ontology Driven Data Mining since the DSO & patient ontologies help clinicians in extracting the right attributes and classes from the raw clinical data. The details of the two complimentary DDx approaches have been introduced in Chapters 3 and 4. The benefits of developing such semantically enriched interactive CDSS model can be summarized as follows:

- **(i)** to influence healthcare providers to reduce variability of outcomes across various health professionals and increase the standardisation of processes towards evidence-based guidelines
- **(ii)** to combine and synthesise complex related pieces of information

⁶⁸ <http://www.wix.com/pjq19007/nursinginformatix>

- (iii) to support the generation of patient-specific (medical history) and context specific prompts and reminders [Bouamrane 2010]
- (iv) to identify patterns within the patient data which must be acted upon (e.g. abnormal or inconsistent findings, alerts, ordering of tests and further investigations, referral to specialist consultant...) [Bouamrane 2010]
- (v) to provide diagnostic recommendations to health care providers

The developed DDx recommender needs to be used by clinicians at various levels of the clinical processes including observation, opinion, instruction and intervention [Kuhn 2007] (see Figure 5.1):

- **Clinical Observation:** information created by an act of observation, measurement, questioning, or testing of the patient's substance (tissue, urine etc), including by the patient himself (e.g. taking own blood glucose measurement), in short, the entire stream of information captured by the investigator, used to characterise the patient system
- **Clinical Opinion:** thoughts of the clinician about what the observations mean, and what to do about them, created during evaluation activities, including all diagnoses, assessments, speculative plans, etc
- **Clinical Instruction:** opinion-based instructions sufficiently detailed so as to be directly executable by investigator agents (people or machines), in order to effect a desired intervention (including obtaining a sample for further investigation, as in a biopsy); and
- **Clinical Intervention:** a record of intervention actions that have occurred, due to clinical instructions or otherwise

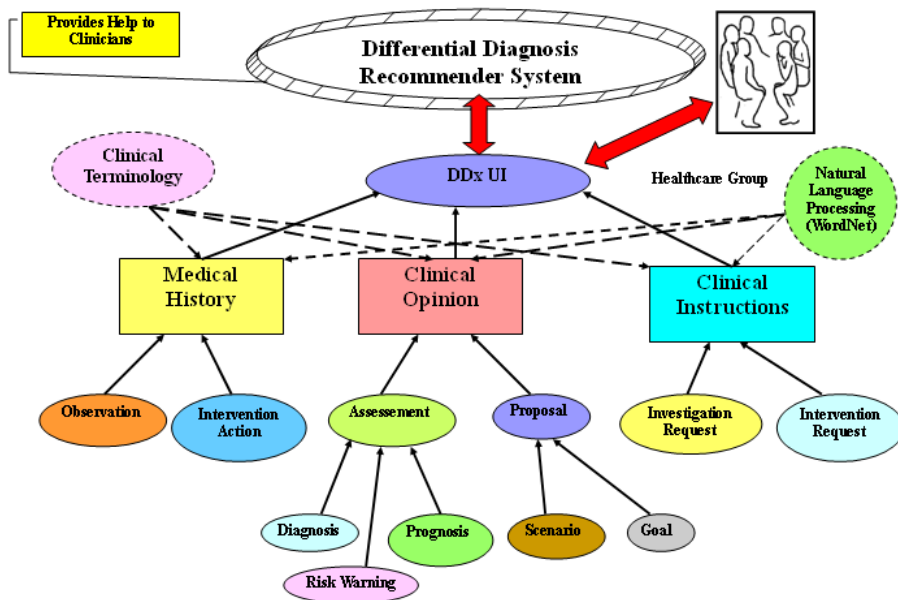


Fig. 5.1: Forms of Interactions with our DDx Recommender

For this purpose the DDx recommender needs to be prototyped in way to enable clinicians to easily make informed diagnostic choices and to be able to query and interact with the DDx recommender in flexible manners.

5.2. Prototyping our Differential Diagnosis Recommender

Prototyping requires rigorous software engineering methodologies that rely on agility. One of the main goals of agile development⁶⁹ is to improve the reaction to changes. Changing user interfaces (UI) is expensive in term of time and resources. Indeed, UI changes do not only impact the UI layer, but also requires several architectural changes. To support developing flexible UI, we have looked into tools satisfying the developer needs (quick development, support all the possible clinical interactions, understandable results) as well as the clinician's needs. For this purpose we chose "Google Web Toolkit (GWT)"⁷⁰ for our DDx wire framing and prototyping. Although wire framing in software industry is a phase used at the early development process, we use it at a later stage when the research concepts of our DDx recommender started to become mature. However, the use of wire framing for DDx recommendation allows us to do many prototyping processes such as:

- Test and refine ontology navigation and crawling
- See how content lays out on the UI pages
- Study and rapidly refine the UI design of forms and interactive elements
- Evaluate overall effectiveness of the page layout against DDx usability best practices
- Determine the integration requirements between the various components and APIs
- Provide the most effective screen workflow

From a developer's point of view, one can easily change the wireframe in response to changes in the requirement. The wireframe produced by the wire framing tool is a natural Java Swing component that can be adjusted easily from the Java project or via the same wire framing toolkit. Figure 5.2 illustrates the DDx recommender's style of screen workflow.

⁶⁹ <http://www.agile-process.org/>

⁷⁰ <http://code.google.com/webtoolkit/>

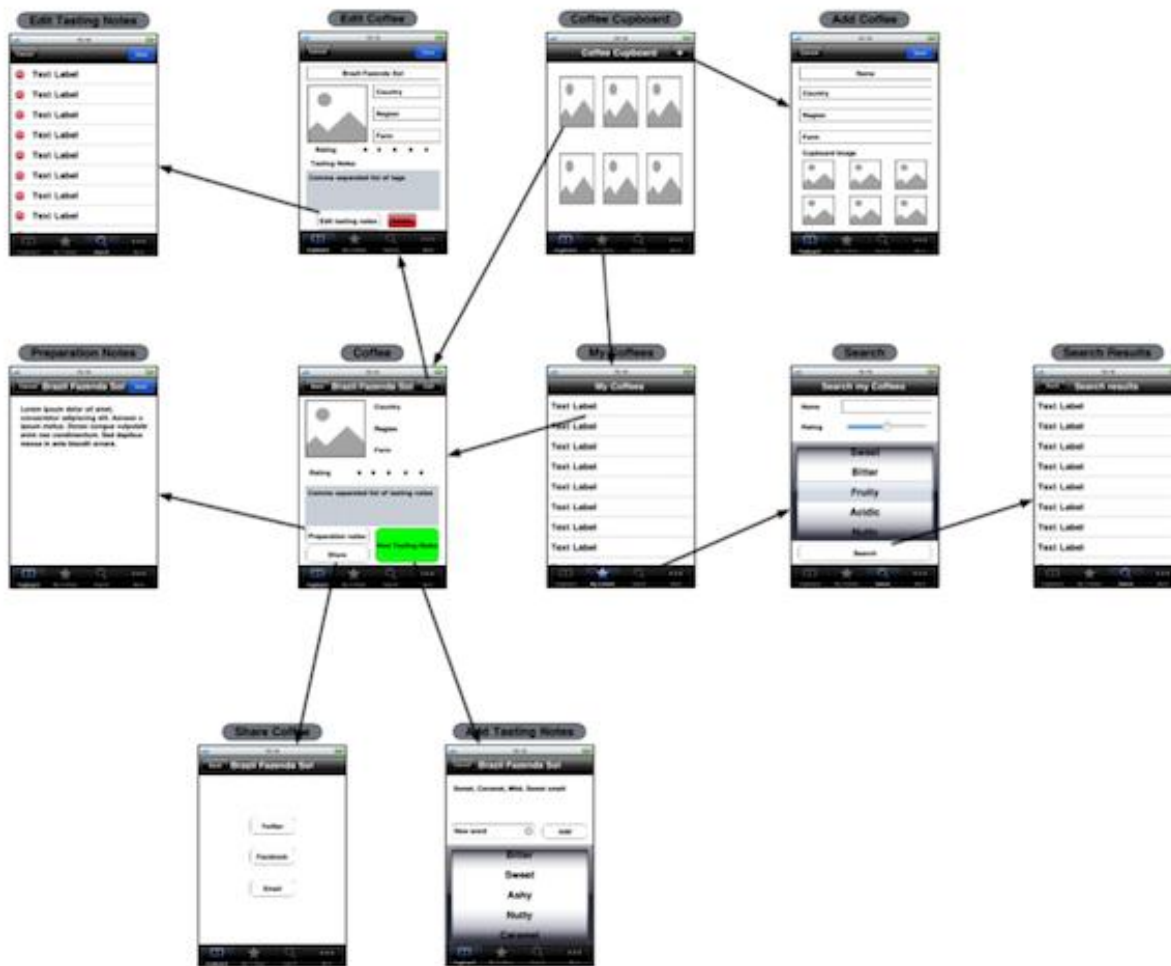


Fig. 5.2: The Style for the DDx Recommender Workflow

5.3. Validation Scenarios for the DDx Recommendation Prototype

This section provides test scenarios of the three main components of the DDx recommendation prototype. Section 5.3.1 provides test scenarios for using the DSO ontology crawler relations R1-R6. Here, we only provide test scenarios for the relations R2, R3, and R5. Relations R1-R6 answer queries relating symptoms to diseases (and vice versa) and are discussed in detail in chapter 3. Section 5.3.2 provides a test scenario for the Evidence-based DDx recommender (chapter 4). Finally, section 5.3.3 presents a test scenario for the Proximity-based DDx recommender (chapter 4).

5.3.1: Validation of the DDx Recommender’s DSO Ontology Crawler Relations

The R1(described in chapter 3) relation is where the DSO Ontology Crawler receives a single symptom as input, and outputs diseases that cause this symptom. Our R1 scenario starts by the

clinician accessing the DSO Ontology Crawler by clicking the "Differential Diagnosis Investigation" button in the main window of the DDx Recommender (Figure 5.3). Then in the next screen, the clinician presses "Investigate R1/R2" button to access the R1 function (Figure 5.4). Then for this scenario, let's suppose the clinician would like to view diseases that cause the symptom fatigue. In this case, the clinician selects the fatigue symptom from a list of symptoms provided by the DSO Crawler. The clinician then presses the "Get Diseases of Symptoms (Text View)" button to get a list of the diseases that cause fatigue (Figure 5.5). The clinician can also get a graph view connecting the symptom to the diseases that cause it by pressing the "Get Diseases of Symptoms (Graph View)". The DDx recommender will display the graph in a separate screen (Figure 5.6).

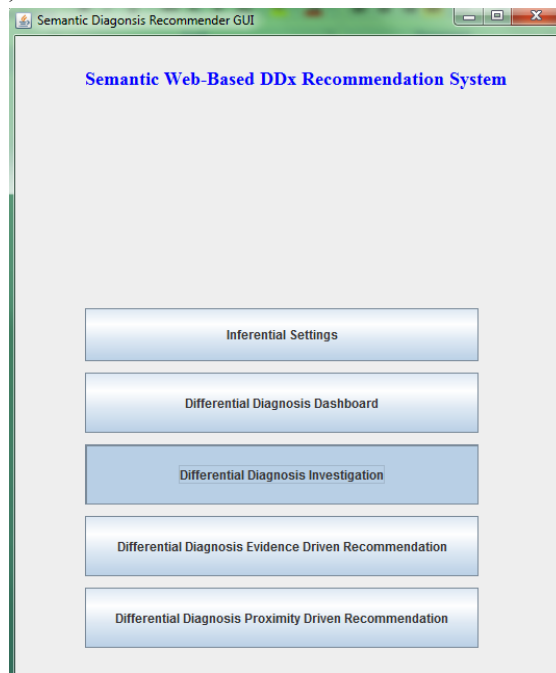


Fig. 5.3: Accessing the DSO Crawler Component of the DDx Recommender

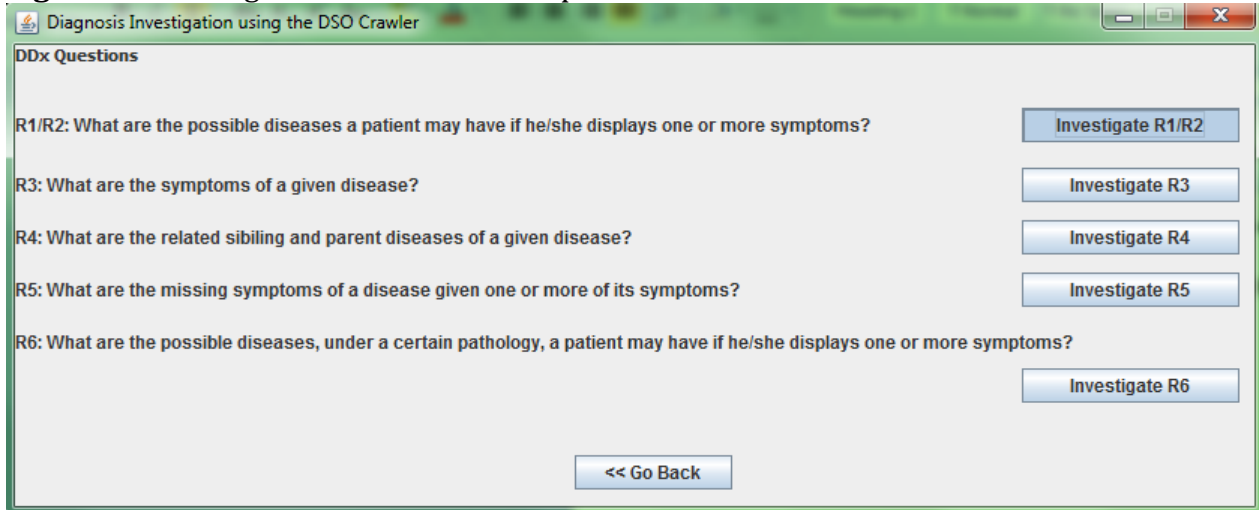


Fig. 5.4: R1-R6 Relations Window of the DSO Crawler

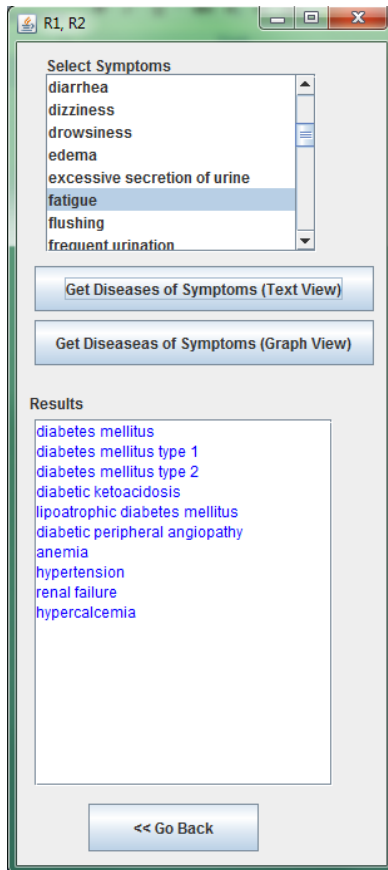


Fig. 5.5: Result of running R1 for the Fatigue Symptom

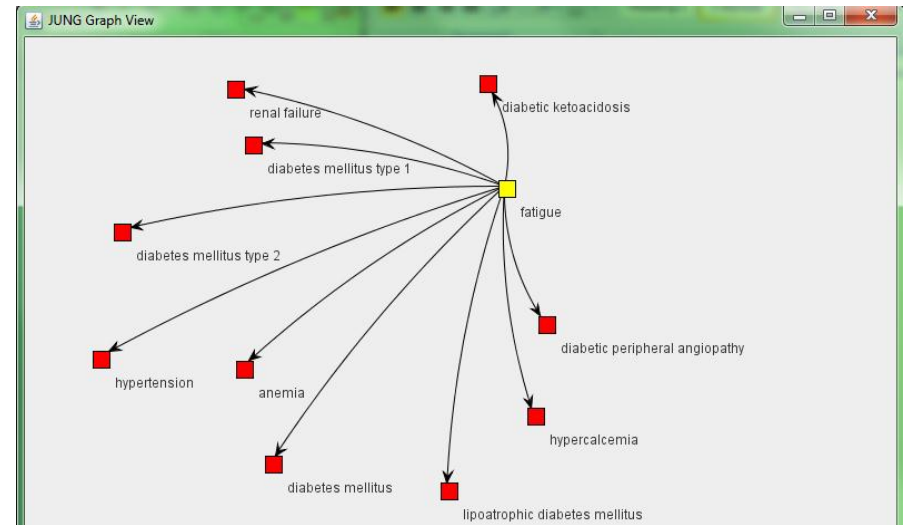


Fig. 5.6: Jung Graph View showing result of running R1 for the Fatigue Symptom

For R2 (chapter 3), the clinician can select a list of symptoms and the DSO Crawler should return a list of diseases where each disease causes all the given symptoms. Figure 5.7 illustrates a scenario where the clinician selects fatigue and headache symptoms and the DSO Crawler returns a list of diseases that cause these symptoms.

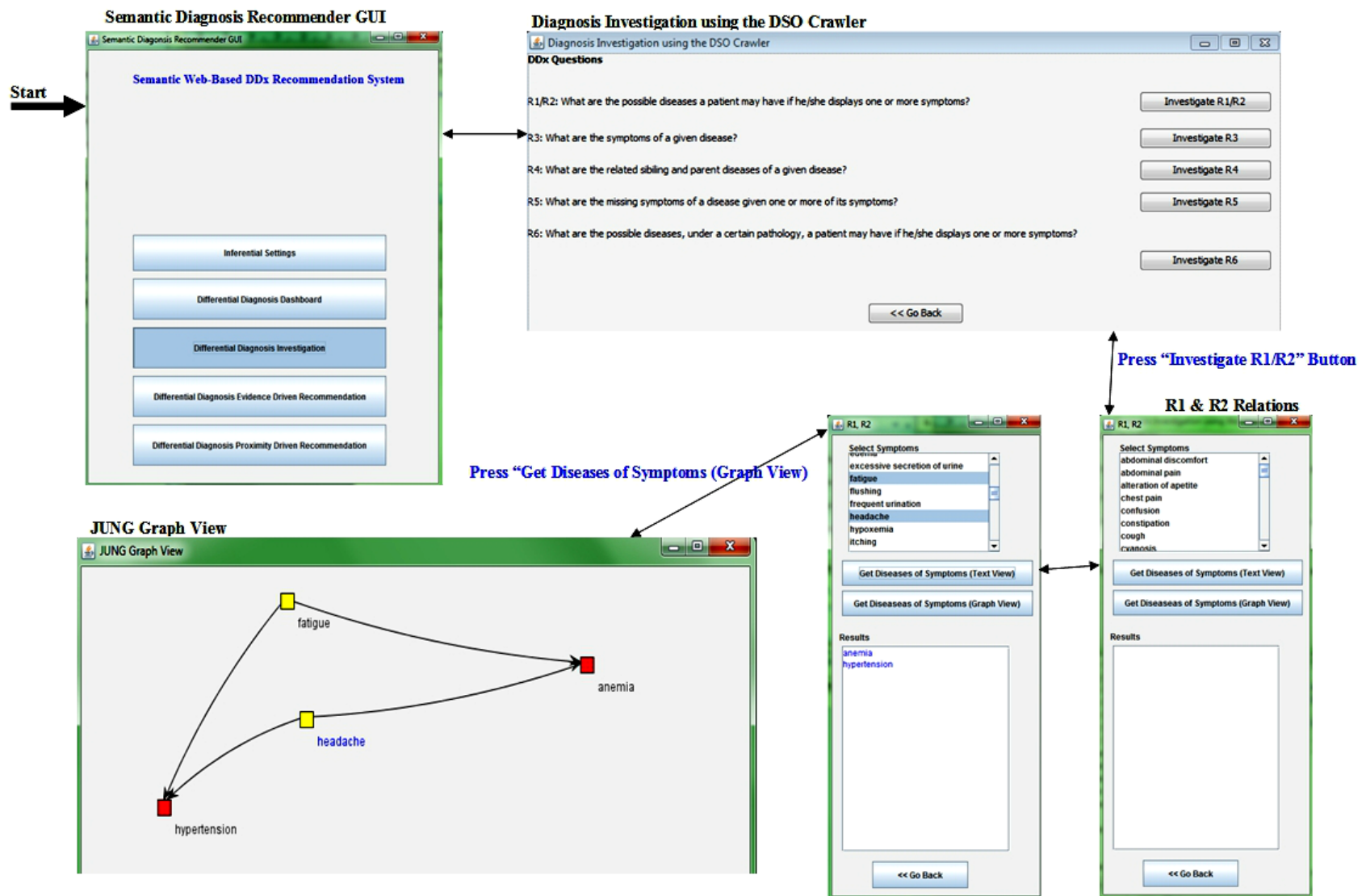


Fig. 5.7: R1/R2 Scenario for the DSO Crawler of the DDX Recommender

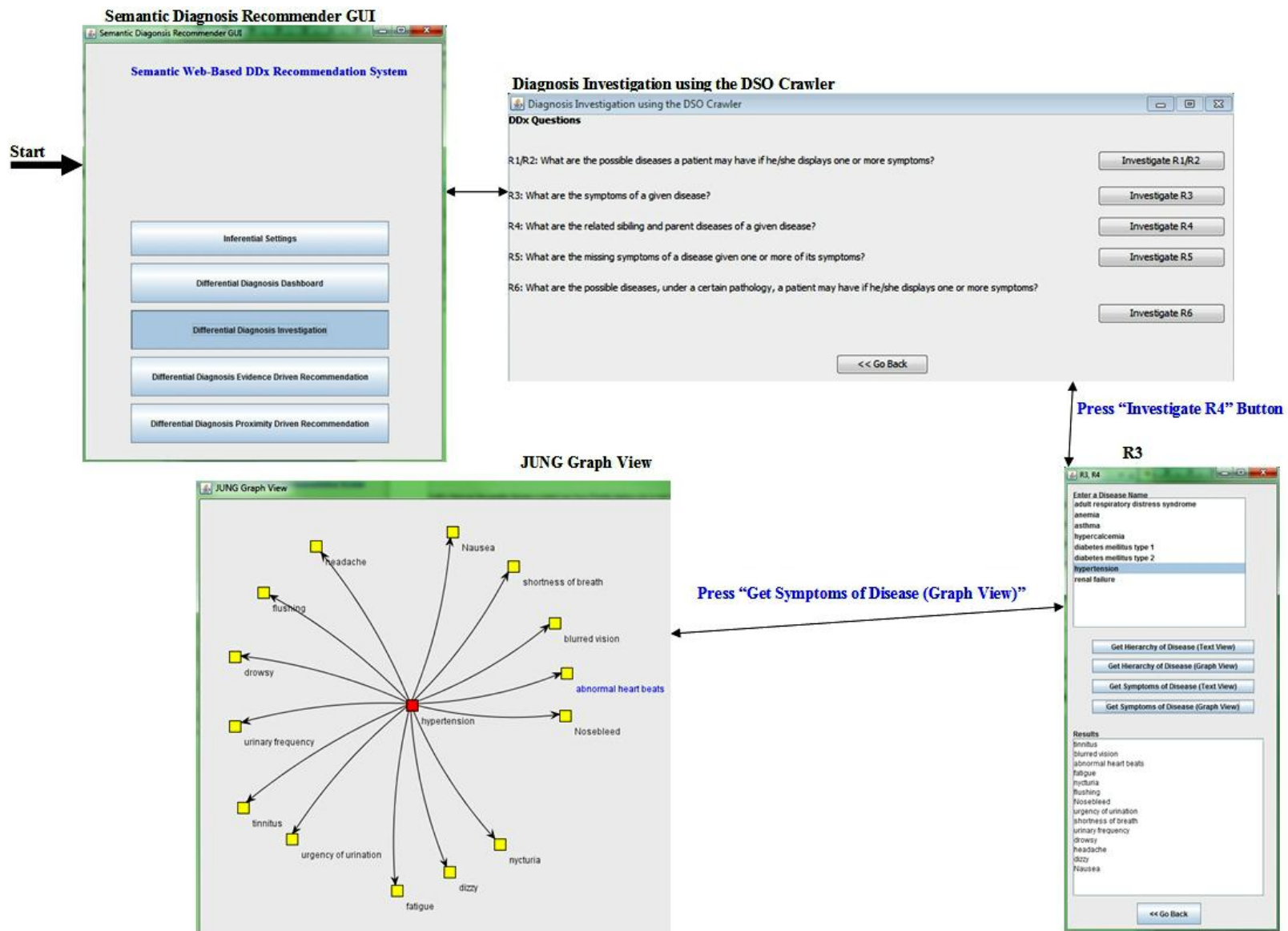


Fig. 5.8: R3 Scenario for the DSO Crawler of the DDx Recommender

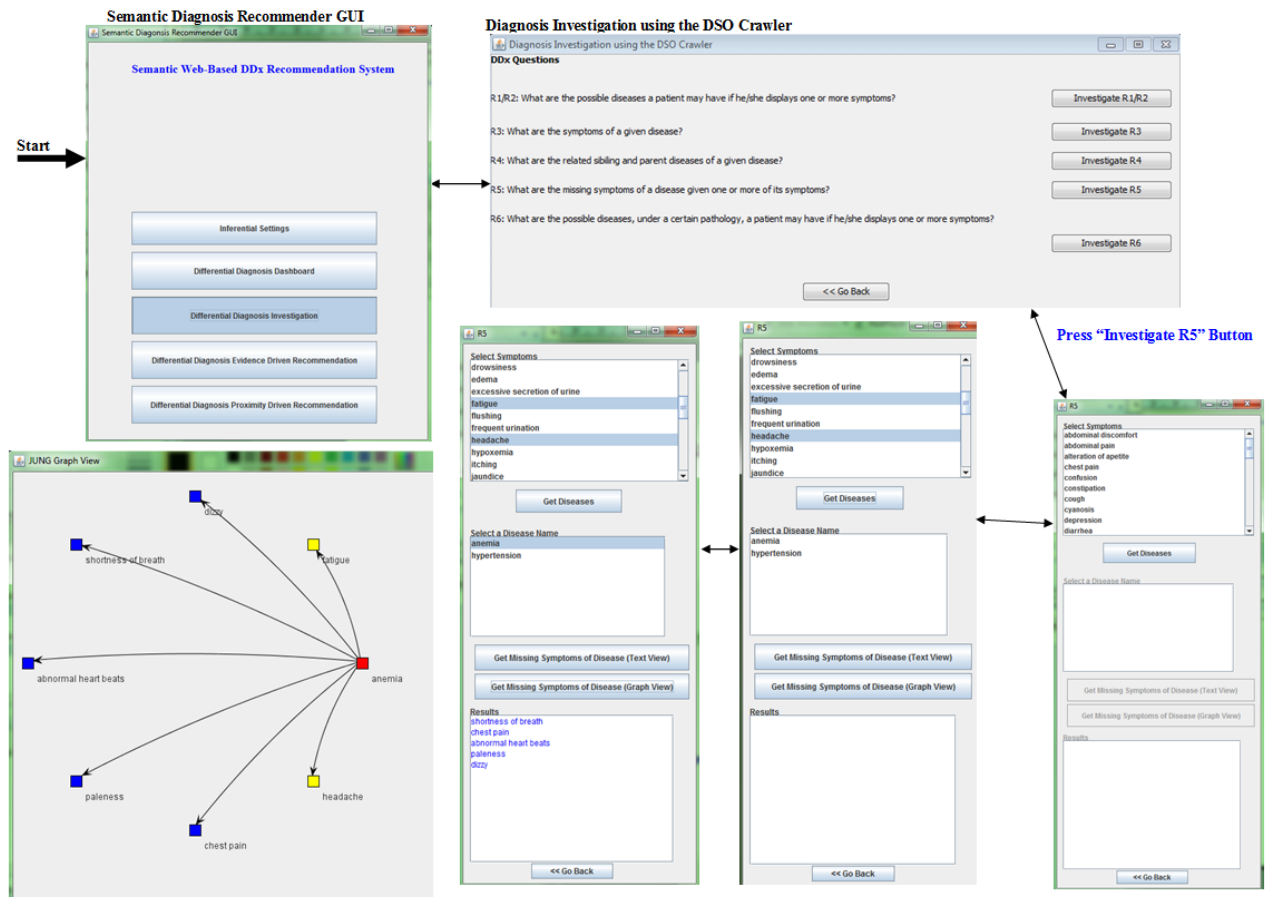


Fig. 5.9: R5 Scenario for the DSO Crawler of the DDx Recommender

5.3.2: Validation Scenario for the Evidence-based DDx Recommender

To run the evidence-based DDx recommender, the clinician needs to press on the "Differential Diagnosis Evidence Driven Recommendation" button in the main window of the DDx recommender (Figure 5.10). In this scenario, a clinician initially observes two symptoms on a patient namely fatigue and a headache. In our prototype, the clinician selects the two symptoms from a list of symptoms (See Figure 5.11 below). Then the clinician presses the "Show Possible Diseases" button to prompt our DDx recommender to show diseases that display the selected symptoms. The DDx recommender, using its DSO crawler described in chapter 3, returns a list of diseases that display the selected symptoms. Among the diseases returned are hypertension and anemia. Let's suppose that the clinician ruled out the remaining diseases that display fatigue and headache as symptoms. Now the clinician would like to look into the possibility that the patient has hypertension. The clinician would take the patient's blood pressure, perhaps several times at different times in a day. As far as our prototype is concerned, the clinician selects hypertension from the list of possible diseases, and presses "Provide Patient Data for Selected Disease" button in order to further examine the possibility of the hypertension diagnosis. The DDx recommender then would present to the clinician a screen that prompts the clinician to enter

blood pressure of the patient (systolic & diastolic blood pressures). The DDx recommender recognizes, through its clinical pathways rules, that blood pressure is the most important indicator of hypertension and that's why it asks the clinician to take the patient's blood pressure and enter the systolic and diastolic blood pressure measurements of the patient (Figure 5.12). Next, the clinician enters the systolic and diastolic blood pressure measurements of the patient and presses the "Narrow Diagnosis" button (Figure 5.13). The DDx recommender takes the blood pressure measurements and feeds them to the Drools rule engine of the evidence-based DDx recommender. The rule engine takes the blood pressure data, and applies it to the clinical pathways rules related to hypertension. Based on these rules in the rule engine's rule base, the blood pressure measurement is determined to be either normal or hypertensive. In this case, both systolic and diastolic blood pressures are determined to be normal (Figure 5.14). The end result is that hypertension is ruled out. Now only anemia remains in the list of possible diseases. For anemia, the main lab test is CBC (Complete Blood Count) involving blood tests. There are two main measures for CBC. The measure is of the percentage of red blood cells in a sample of blood. The second one is hemoglobin levels and is a measure of hemoglobin levels in a sample of blood. The clinician must order the CBC test in order to investigate diagnosis of anemia. Once the results come in, the clinician can use the aid of the DDx recommender in the diagnosis of anemia. The clinician must select anemia in the list of possible disease and "Provide Patient Data for Selected Disease" button in order to further examine the possibility of the anemia diagnosis (Figure 5.15). Then, the DDx recommender presents the diagnosis rules for the diagnosis of anemia (Figure 5.16). Next, the clinician enters the hemoglobin levels and percentage of red blood cells in the blood sample of the patient. The clinician then presses the "Narrow Diagnosis" button to ask the DDx recommender to investigate the possible diagnosis of anemia (Figure 5.17). When it comes to interpreting the test results of the CBC test, normal and abnormal results differ depending on whether the patient is male or female. Therefore, the DDx recommender will ask the clinician for the patient's sex as well as age. The clinician then would enter the patient's sex and age (Figure 5.18). The DDx recommender takes the CBC lab test measurements and feeds them to the Drools rule engine of the evidence-based DDx recommender. The rule engine takes the hemoglobin and red blood cell percentage data, and applies it to the clinical pathways rules related to anemia. Based on these rules in the rule engine's rule base and the CBC test results in this case, the DDx recommender determines that the patient has anemia. Therefore, the DDx recommender suggests a diagnosis of anemia to the patient (Figure 5.19). Since only one disease (Anemia) remains in the list of possible diseases, the DDx recommender has no more diseases to investigate. The last step the DDx recommender performs in this diagnosis sequence is part of the cooperation between the evidence-based and proximity-based DDx recommenders. The above process was controlled by the evidence-based DDx recommender. As part of the cooperation between both recommenders, the evidence-based recommender will write its findings to the proximity-based recommender. For each of disease recognized by the DDx recommender, the proximity-based recommender maintains records of patient data where a

diagnosis is recorded based on the evidence-based recommender's conclusion. In figure 5.20, the evidence-based recommender writes the diagnosis of anemia as a patient record in a table of records for anemia where the diagnosis is either confirmed anemia or no indication of anemia (normal). The last record in figure 5.20 belongs to the patient's case we are discussing in this scenario. It shows the patient's age in years (25.0), age in days (0.0), sex (1.0) where 0.0 stands for male and 1.0 stands for female, the percentage of red blood cells in patient's blood sample (33), the hemoglobin levels in the blood sample (11), and a positive diagnosis of anemia (Anemia). This scenario would also add a record to the hypertension table of the proximity-based recommender where the diagnosis is "Normal" or no indication of hypertension (Figure 5.21). The last record in figure 5.21 is the record added in this scenario, It shows the patient's age in years (23.0), age in day (0.0), sex(1.0 or female), systolic blood pressure (123), diastolic blood pressure (78), and the diagnosis of either normal or hypertension (Normal).



Fig. 5.10: Selection of the Evidence-based DDX Recommender from the Main GUI Window of the DDX Recommender

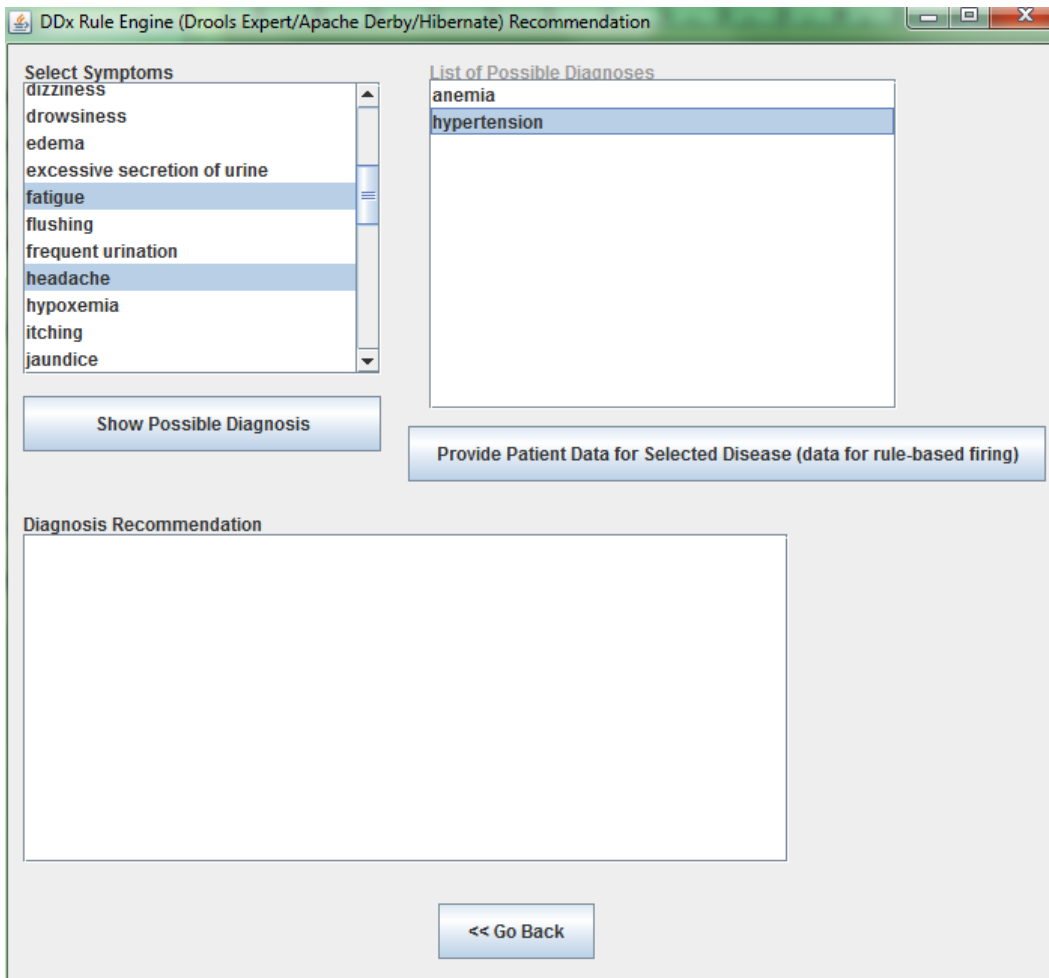


Fig. 5.11: Main GUI Window of the Evidence-based DDx Recommender

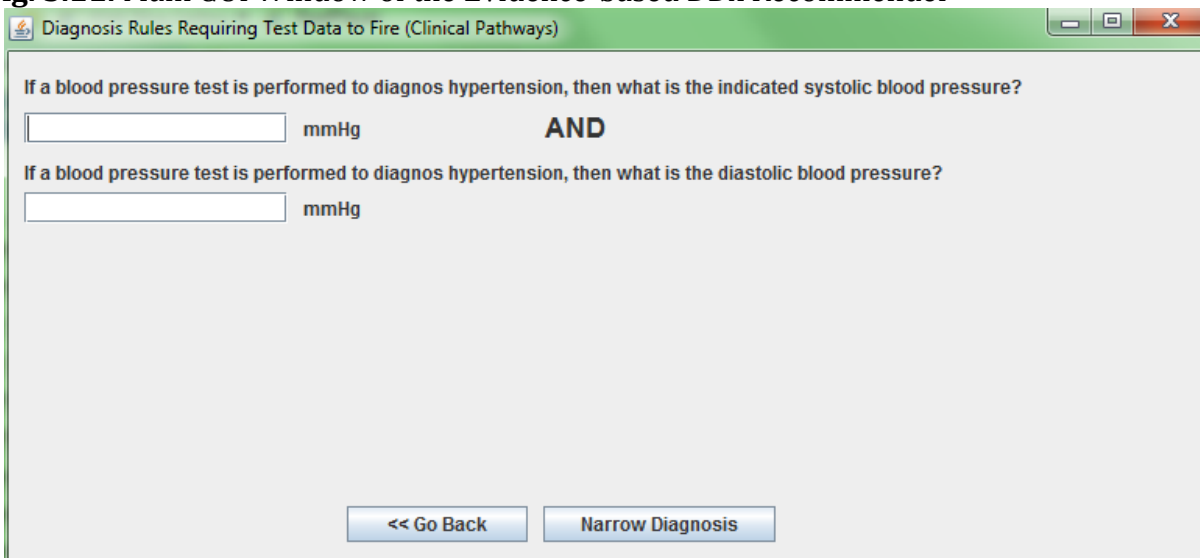


Fig. 5.12: DDx Recommender's GUI Window for Clinical Pathways Rules for Hypertension

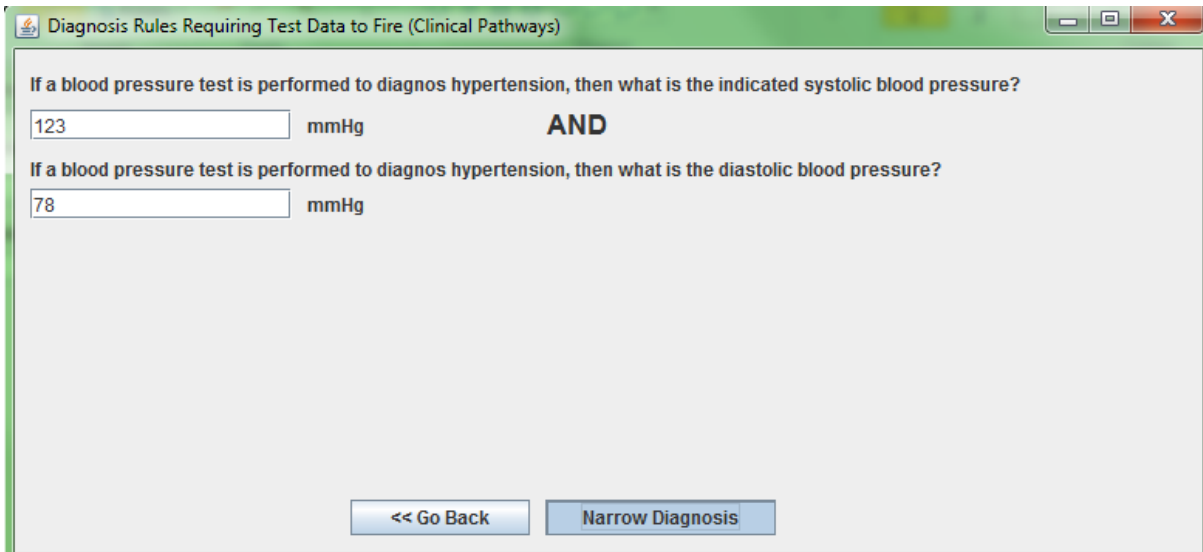


Fig. 5.13: Clinician Enters Blood Pressure of Patient into DDx Recommender

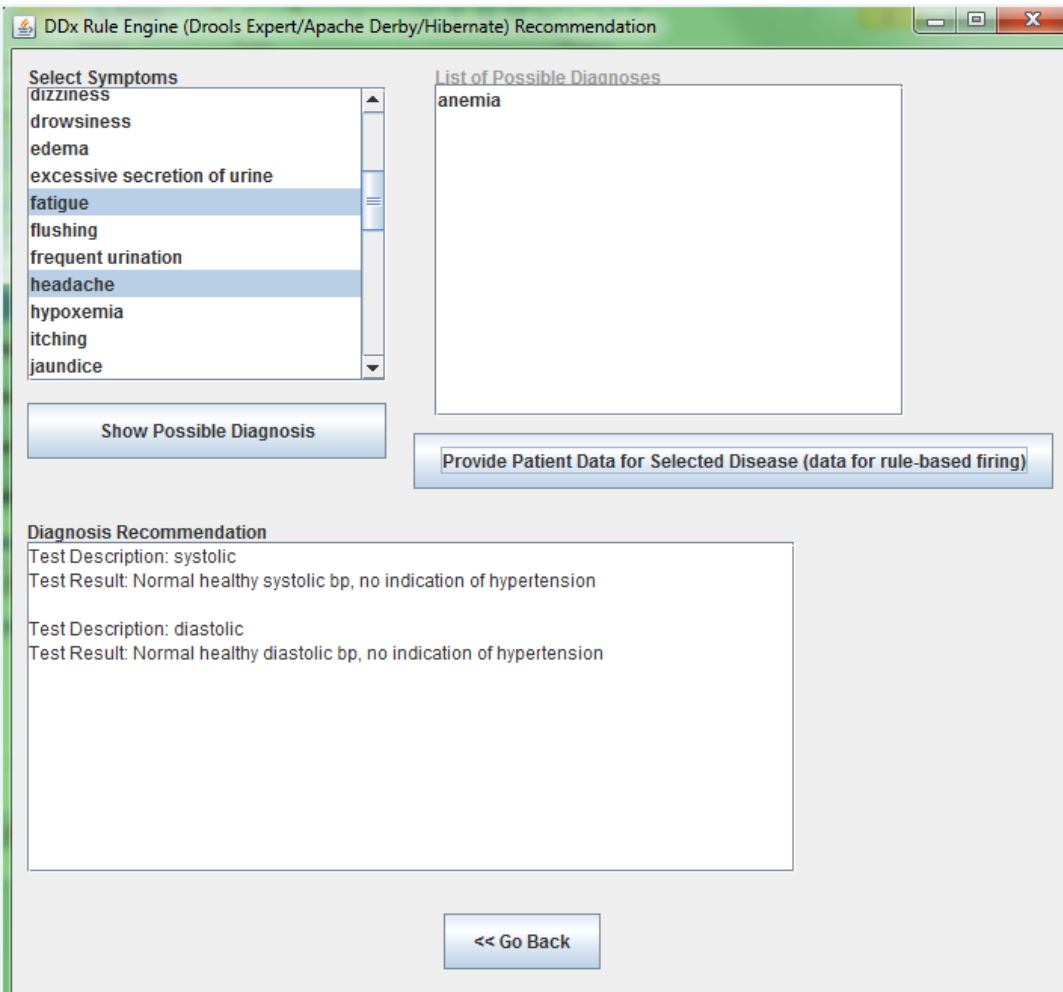


Fig. 5.14: DDx Recommender Rules Out Hypertension

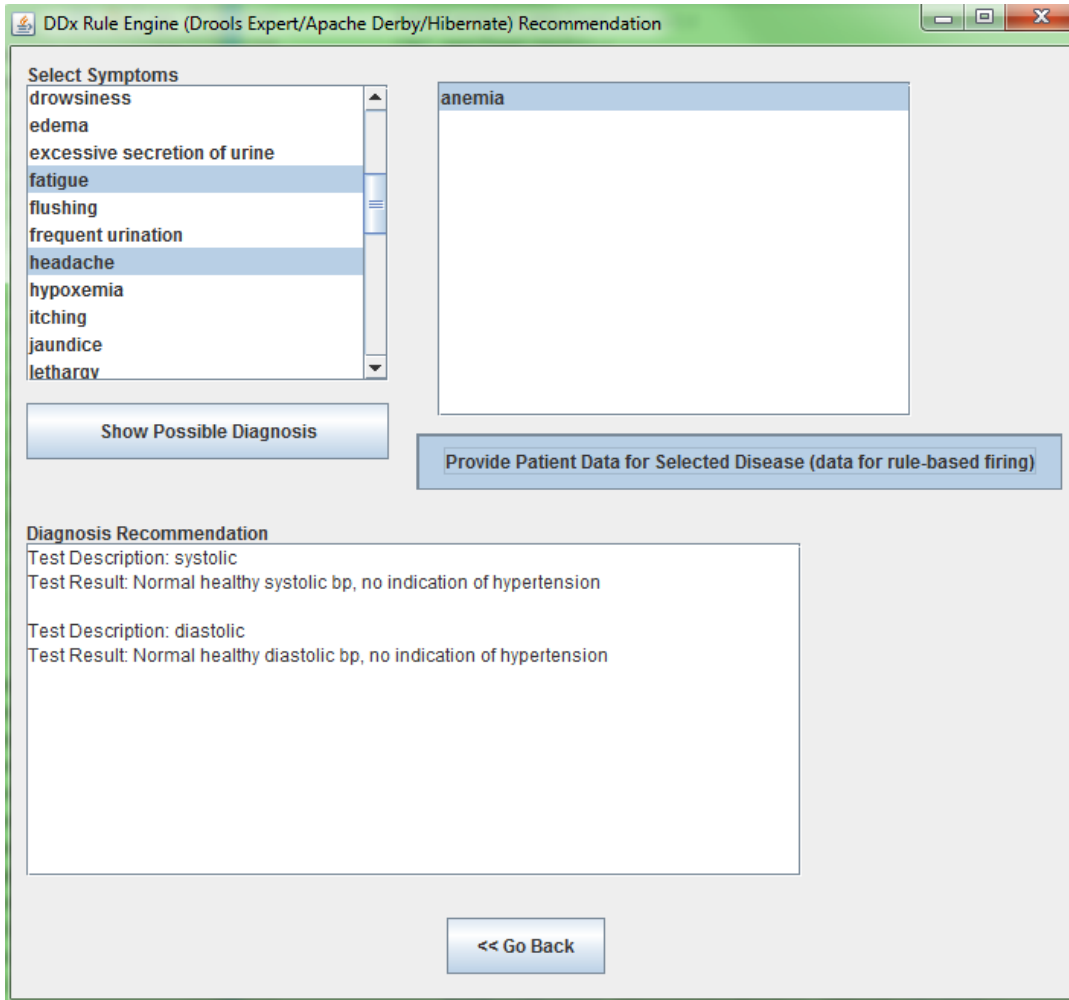


Fig. 5.15: Clinician Using DDX Recommender to Examine the Possible Diagnosis of Anemia

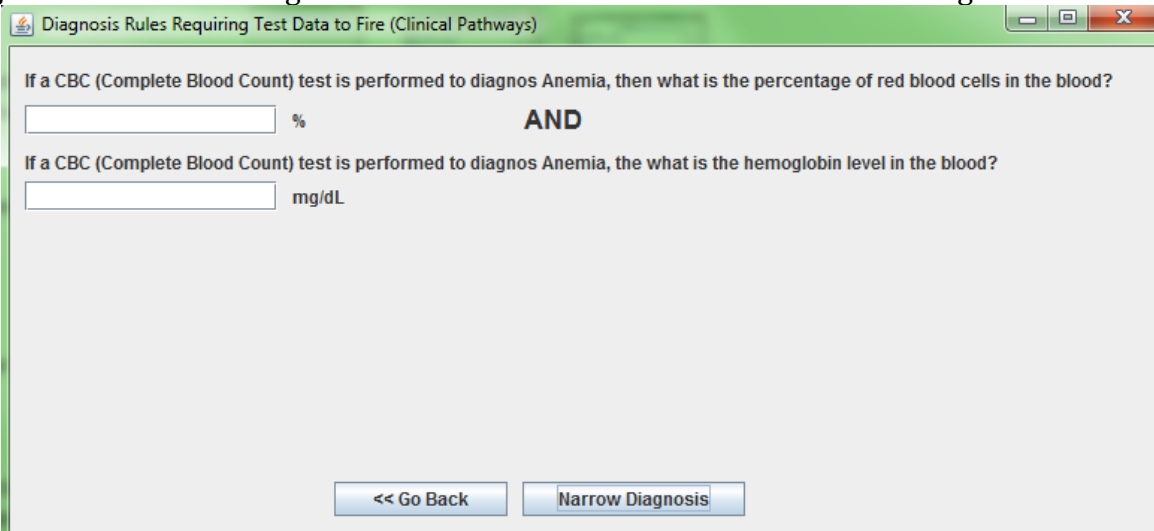


Fig. 5.16: DDX Recommender's Clinical Pathways Rules for Anemia

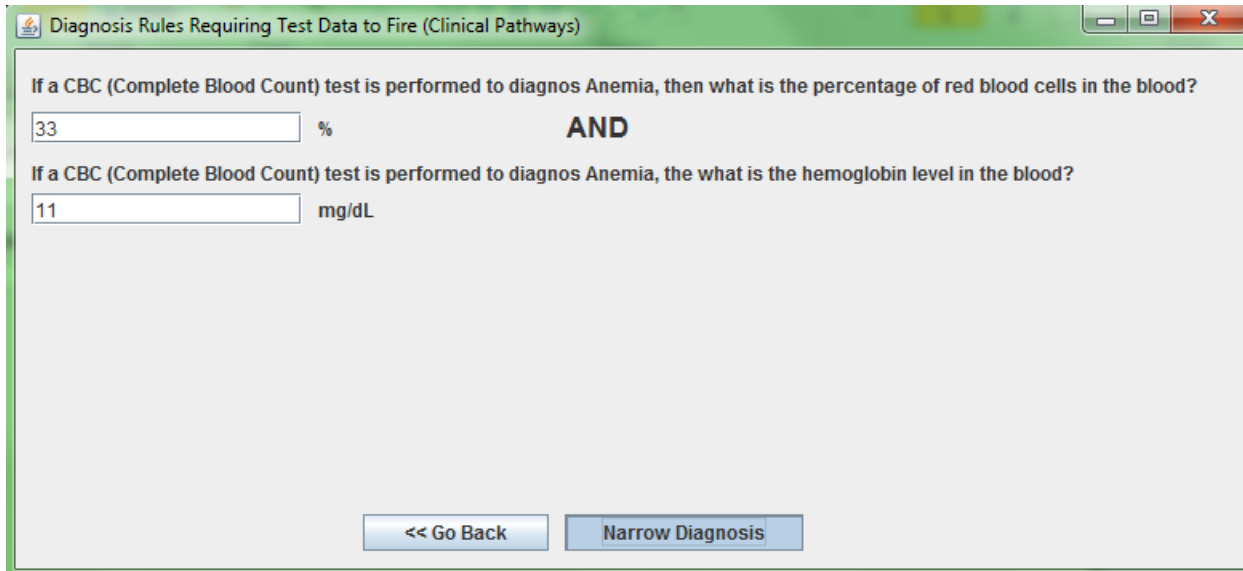


Fig. 5.17: Clinician Enters Test Data for Anemia and Prompts DDx Recommender to Fire its Clinical Pathways Rules

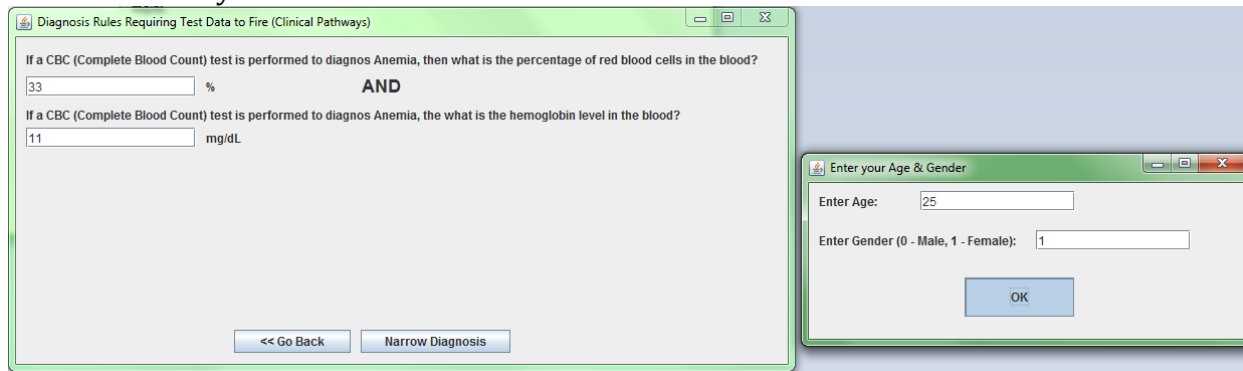


Fig. 5.18: DDx Recommender Prompts Clinician to Enter Patient's Sex and Age

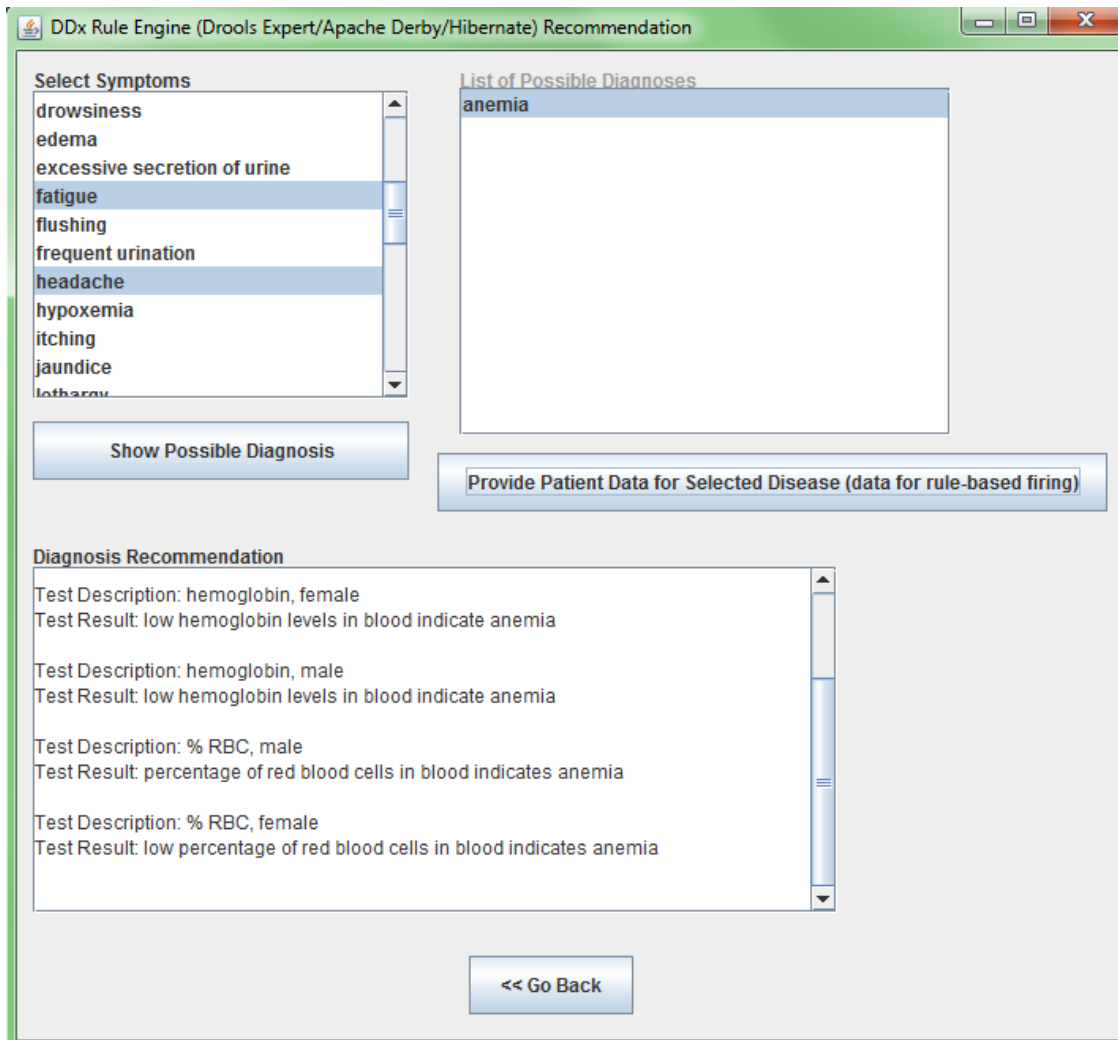


Fig. 5.19: DDx Recommender Suggests to Clinician a Diagnosis of Anemia

84.0,	0.0,	1.0,	26.88796348732096,	9.267203054738681,	"ANEMIA NOS--"
84.0,	0.0,	1.0,	40.250845419148206,	14.919722953815574,	"Normal"
67.0,	0.0,	0.0,	34.529526271920545,	12.685495123167868,	"ANEMIA NOS--"
72.0,	0.0,	0.0,	43.189163842208615,	13.940547661170228,	"Normal"
64.0,	0.0,	1.0,	29.091940979956863,	8.753780220274727,	"ANEMIA NOS--"
49.0,	0.0,	1.0,	40.66954484227705,	13.281540249999262,	"Normal"
44.0,	0.0,	1.0,	32.28296697750094,	10.334083674892955,	"ANEMIA NOS--"
45.0,	0.0,	0.0,	47.01261756152666,	14.507982557679533,	"Normal"
5.0,	0.0,	1.0,	34.5943832013846,	8.30051552029945,	"ANEMIA NOS--"
48.0,	0.0,	0.0,	44.317440489578416,	13.706040190110095,	"Normal"
47.0,	0.0,	0.0,	33.03722737857008,	10.053362810748563,	"ANEMIA NOS--"
50.0,	0.0,	0.0,	46.359861834830006,	16.965416300004925,	"Normal"
91.0,	0.0,	1.0,	33.73414127117614,	11.494424135702388,	"ANEMIA NOS--"
62.0,	0.0,	0.0,	44.845981228485535,	14.912282036222601,	"Normal"
35.0,	0.0,	0.0,	30.366661176407597,	13.490354000099522,	"ANEMIA NOS--"
26.0,	0.0,	1.0,	35.94714068241476,	14.985803344642036,	"Normal"
38.0,	0.0,	0.0,	36.3019648121594,	13.137299656958685,	"ANEMIA NOS--"
0.0,	0.0,	0.0,	48.68749615563769,	13.55858166460474,	"Normal"
0.0,	71.0,	0.0,	32.338129681421165,	10.648657887529609,	"ANEMIA NOS--"
1.0,	0.0,	0.0,	42.98387747000203,	14.39232380903335,	"Normal"
11.0,	0.0,	1.0,	27.960799988565075,	7.793457988592657,	"ANEMIA NOS--"
0.0,	94.0,	1.0,	42.18509142382119,	12.98332735653507,	"Normal"
0.0,	49.0,	0.0,	38.194454057510114,	13.415325942858487,	"ANEMIA NOS--"
11.0,	0.0,	1.0,	36.225074533836164,	13.842860126904384,	"Normal"
0.0,	65.0,	0.0,	32.014340817677635,	10.070387849383692,	"ANEMIA NOS--"
8.0,	0.0,	0.0,	45.20323283488964,	14.93537231841246,	"Normal"
17.0,	0.0,	1.0,	33.775079731069454,	10.108270464357851,	"ANEMIA NOS--"
1.0,	0.0,	0.0,	46.006550792806586,	14.72123095704115,	"Normal"
95.0,	0.0,	1.0,	34.7544573523554,	11.389460421435551,	"ANEMIA NOS--"
68.0,	0.0,	1.0,	37.00939908900572,	12.946501252574263,	"Normal"
83.0,	0.0,	1.0,	28.37099453269892,	10.56861687050072,	"ANEMIA NOS--"
59.0,	0.0,	1.0,	41.40059461272852,	14.957708747486938,	"Normal"
74.0,	0.0,	0.0,	33.38796297550207,	9.298506368456623,	"ANEMIA NOS--"
50.0,	0.0,	0.0,	42.273337677108884,	14.972399061658214,	"Normal"
91.0,	0.0,	1.0,	32.951316219444756,	11.862225167048248,	"ANEMIA NOS--"
63.0,	0.0,	0.0,	41.66304502487231,	13.82969283041157,	"Normal"
77.0,	0.0,	1.0,	25.82297394052701,	9.141165439702128,	"ANEMIA NOS--"
78.0,	0.0,	1.0,	37.88949958786162,	13.438215923871947,	"Normal"
87.0,	0.0,	1.0,	27.748330307925887,	10.247384176860947,	"ANEMIA NOS--"
23.0,	0.0,	1.0,	41.337295986724996,	15.415132893506842,	"Normal"
88.0,	0.0,	1.0,	28.77456813739127,	10.270048259672093,	"ANEMIA NOS--"
40.0,	0.0,	1.0,	39.27791958723249,	12.893938428827004,	"Normal"
62.0,	0.0,	1.0,	33.17668149284742,	10.356834791779294,	"ANEMIA NOS--"
0.0,	290.0,	1.0,	36.316503038430064,	13.118341463772202,	"Normal"
75.0,	0.0,	1.0,	32.607518822422435,	10.55407352116569,	"ANEMIA NOS--"
82.0,	0.0,	0.0,	44.641885148813635,	16.43101977694245,	"Normal"
42.0,	0.0,	1.0,	28.67525132837369,	11.961635876730167,	"ANEMIA NOS--"
44.0,	0.0,	0.0,	42.67251001685412,	16.095024945245022,	"Normal"
30.0,	0.0,	0.0,	40, 14,	"Normal"	
25.0,	0.0,	1.0,	33, 11,	"ANEMIA NOS--"	
23.0,	0.0,	1.0,	33, 11,	"ANEMIA NOS--"	

Fig. 5.20: Anemia Patient Record Table of the Proximity-based DDx Recommender

29.0,	0.0,	1.0,	86,	44,	"ORTHOSTATIC HYPOTENSION"
72.0,	0.0,	1.0,	69,	51,	"ORTHOSTATIC HYPOTENSION"
68.0,	0.0,	1.0,	87,	53,	"ORTHOSTATIC HYPOTENSION"
49.0,	0.0,	1.0,	158,	96,	"HYPERTENSION NOS"
57.0,	0.0,	0.0,	60,	55,	"ORTHOSTATIC HYPOTENSION"
67.0,	0.0,	1.0,	160,	103,	"HYPERTENSION NOS"
34.0,	0.0,	1.0,	162,	100,	"HYPERTENSION NOS"
67.0,	0.0,	1.0,	141,	117,	"HYPERTENSION NOS"
47.0,	0.0,	0.0,	66,	55,	"ORTHOSTATIC HYPOTENSION"
53.0,	0.0,	0.0,	164,	111,	"HYPERTENSION NOS"
45.0,	0.0,	1.0,	60,	41,	"ORTHOSTATIC HYPOTENSION"
76.0,	0.0,	1.0,	166,	95,	"HYPERTENSION NOS"
81.0,	0.0,	1.0,	143,	104,	"HYPERTENSION NOS"
76.0,	0.0,	1.0,	150,	97,	"HYPERTENSION NOS"
41.0,	0.0,	1.0,	161,	102,	"HYPERTENSION NOS"
54.0,	0.0,	0.0,	153,	98,	"HYPERTENSION NOS"
41.0,	0.0,	0.0,	157,	109,	"HYPERTENSION NOS"
68.0,	0.0,	0.0,	163,	104,	"HYPERTENSION NOS"
58.0,	0.0,	0.0,	158,	101,	"HYPERTENSION NOS"
87.0,	0.0,	1.0,	151,	109,	"HYPERTENSION NOS"
40.0,	0.0,	1.0,	158,	115,	"HYPERTENSION NOS"
93.0,	0.0,	1.0,	157,	114,	"HYPERTENSION NOS"
40.0,	0.0,	0.0,	164,	96,	"HYPERTENSION NOS"
37.0,	0.0,	1.0,	155,	119,	"HYPERTENSION NOS"
82.0,	0.0,	0.0,	75,	58,	"ORTHOSTATIC HYPOTENSION"
82.0,	0.0,	1.0,	63,	47,	"ORTHOSTATIC HYPOTENSION"
50.0,	0.0,	0.0,	162,	99,	"HYPERTENSION NOS"
77.0,	0.0,	1.0,	88,	41,	"ORTHOSTATIC HYPOTENSION"
69.0,	0.0,	0.0,	153,	96,	"HYPERTENSION NOS"
50.0,	0.0,	1.0,	60,	49,	"ORTHOSTATIC HYPOTENSION"
83.0,	0.0,	1.0,	69,	56,	"ORTHOSTATIC HYPOTENSION"
33.0,	0.0,	0.0,	66,	58,	"ORTHOSTATIC HYPOTENSION"
71.0,	0.0,	1.0,	156,	101,	"HYPERTENSION NOS"
57.0,	0.0,	1.0,	148,	114,	"HYPERTENSION NOS"
74.0,	0.0,	0.0,	72,	40,	"ORTHOSTATIC HYPOTENSION"
62.0,	0.0,	1.0,	142,	103,	"HYPERTENSION NOS"
38.0,	0.0,	0.0,	141,	108,	"HYPERTENSION NOS"
51.0,	0.0,	1.0,	157,	111,	"HYPERTENSION NOS"
81.0,	0.0,	1.0,	142,	99,	"HYPERTENSION NOS"
53.0,	0.0,	1.0,	167,	94,	"HYPERTENSION NOS"
62.0,	0.0,	1.0,	147,	91,	"HYPERTENSION NOS"
61.0,	0.0,	1.0,	157,	95,	"HYPERTENSION NOS"
25.0,	0.0,	1.0,	110,	75,	"Normal"
32.0,	0.0,	0.0,	114,	82,	"Normal"
35.0,	0.0,	1.0,	98,	68,	"Normal"
47.0,	0.0,	0.0,	104,	79,	"Normal"
27.0,	0.0,	0.0,	101,	83,	"Normal"
37.0,	0.0,	1.0,	111,	64,	"Normal"
23.0,	0.0,	1.0,	132,	89,	"Normal"
24.0,	0.0,	1.0,	123,	89,	"Normal"
23.0,	0.0,	1.0,	123,	78,	"Normal"

Fig. 5.21: Hypertension Patient Record Table of the Proximity-based DDX Recommender

5.3.3. Validation of the Proximity-based DDX Recommender

To start using the proximity-based DDX recommender, the clinician needs to press the "Differential Diagnosis Proximity Driven Recommendation" button (Figure 5.12). The next screen would appear which gives the clinician two choices: making diagnosis recommendations for patient cases based on known similar patient cases (First choice) or finding diagnosis rules relating various clinical patient attributes and lab data on one hand, and diagnosis of a certain disease on the other hand (Second choice). These two choices are discussed in detail in chapter 4. The first choice relies on classification and is accessed by pressing the "Answer D1" button

(Figure 5.22) where D1 is a relation implementing the first choice. D1 evaluates patient cases based previous similar patient cases. The second choice relies on association and is accessed by pressing the "Answer D2" button (Figure 5.22) where D2 is a relation implementing the second choice. D2 produces diagnosis rules, based on known patient cases, relating clinical data to the diagnosis of a certain disease.

First D1 Scenario

For our first scenario, we will test a D1 Scenario. The clinician presses the "Answer D1" button. The next screen prompts the clinician to select one of four classification algorithms (discussed in chapter 4) to perform D1 (Figure 5.23). Let's say the clinician selects the J48 algorithm in this scenario and presses the "Proceed >>" button (Figure 5.24). The next screen (Figure 5.25) asks the clinician whether he/she would like to view the diagnosis prediction trends learned by the classification algorithm from recorded patient cases where patients were examined for a certain disease and determined to either have or not have the disease (option 1), or get diagnosis predictions for new patient cases based on the recorded patient cases (option 2). Let's say the clinician would like to proceed with option 1. The clinician would select the "Use Training Data to Find Trends in the Data" option and presses the "Proceed >>" button (Figure 5.25). The next (Figure 5.26) screen allows the clinician to select the disease for which to display diagnosis trends. Currently, we have known patient case data for four diseases (Diabetes, Anemia, Calcemia, and Hypertension). The clinician can select any of the listed data tables (Figure 5.26), and let's suppose in this scenario he/she selects diabetes data and clicks on "Proceed To Results" button. Then, the DDx recommender applies the selected data mining classification algorithm to the selected data table to find trends in the data. Next, the DDx recommender displays the results of classification namely classification statistics (Figure 5.27), and a description of the resulting classification tree (Figure 5.28). Interpreting the classification statistics was discussed in chapter 4 while interpreting the classification tree is straightforward. The classification tree shows that if the fasting plasma glucose levels in the blood are less than 89.5 mg/dL, then glucose levels in the blood are normal. Otherwise, the glucose levels are too high and indicate diabetes.

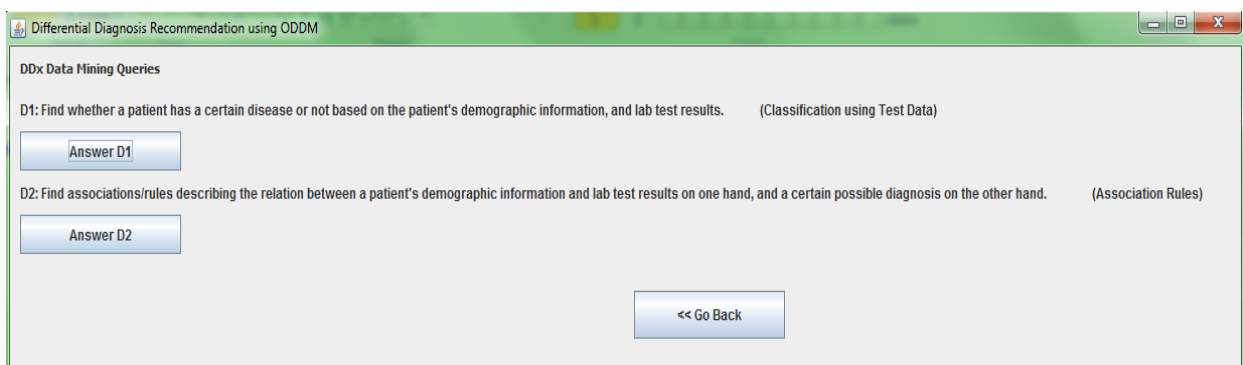


Fig. 5.22: Main GUI Window of the Proximity-based DDx Recommender

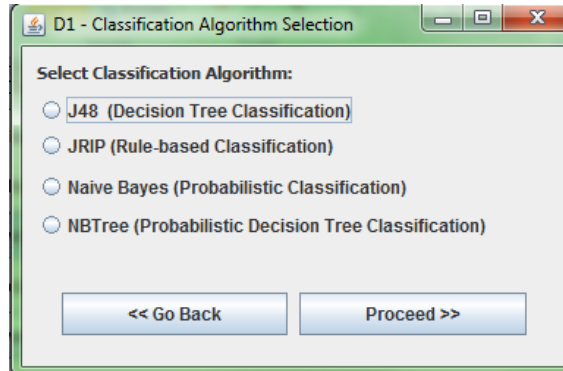


Fig. 5.23: Classification Algorithm Selection Window of the Proximity-based DDx Recommender

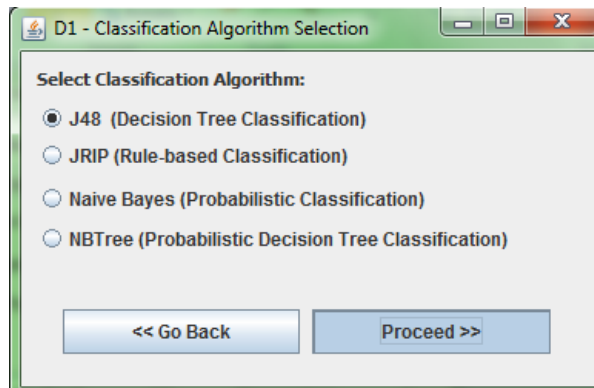


Fig. 5.24: J48 Classification Algorithm selected to perform Proximity-based DDx Recommendation

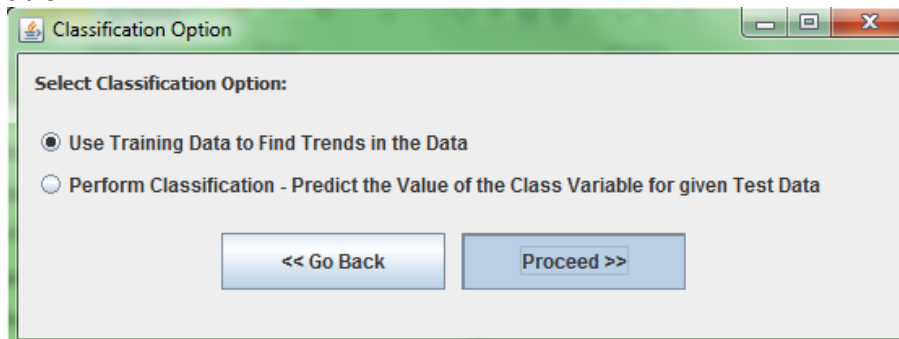


Fig. 5.25: Selecting First Classification Option for the D1 Relation of the Proximity-based DDx Recommender

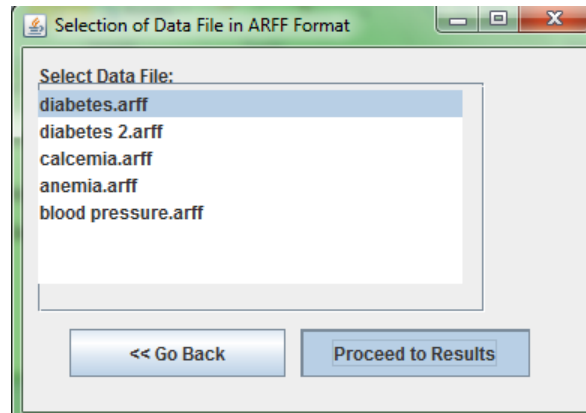


Fig. 5.26: List of Data Tables for the Proximity-based Recommender

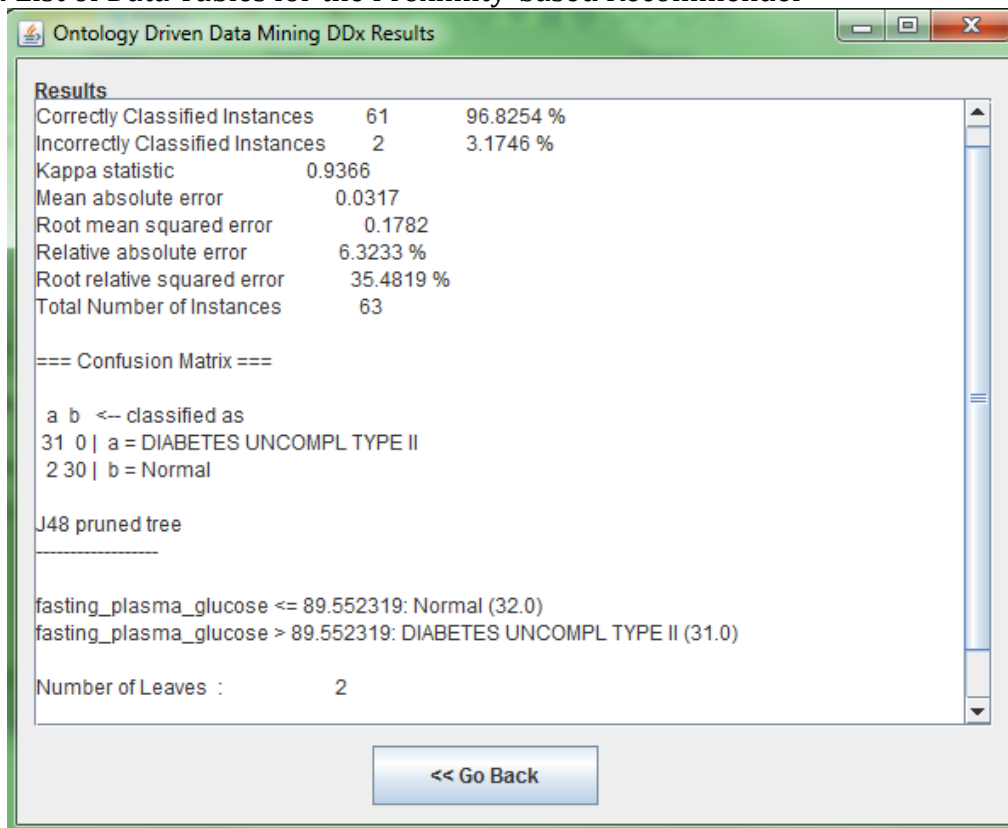


Fig. 5.27: Classification Statistics & J48 Tree for Diabetes of the Proximity-based DDx Recommender

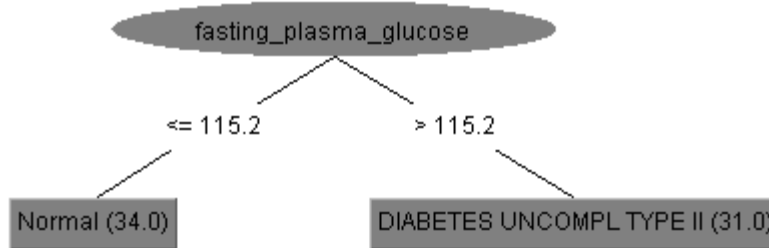


Fig. 5.28: J48 Classification Tree for Diabetes Table of the Proximity-based DDx Recommender

Second D1 Scenario

In this scenario, the clinician is not interested in viewing diagnosis trends obtained from previously known patient cases (option 1), but instead would like to get diagnosis predictions for new patient cases based on the previously known patient cases (option 2). To do so, the clinician proceeds the same way as in the first scenario until the classification options window is reached (Figure 5.29). When the clinician gets to that window, he selects the second option namely "Perform Classification - Predict the Value of the Class Variable for given Test Data" and clicks on the "Proceed >>" button (Figure 5.29). Next, the DDx recommender, for each of the four diseases, has a number of test data records on file meaning a number of diagnosed patient cases but the diagnosis is not recorded on file and therefore unknown to the DDx recommender. The DDx recommender can use the diagnosis trends learned from previously diagnosed patient cases, where the diagnosis is known by the DDx recommender to predict the diagnosis of the test records. The accuracy of prediction is determined by the number of predictions for the test records where the prediction matches the actual diagnosis for the test record. It also allows the clinician to add to new records to the test records. First, the clinician must select the test file with test records for a specific disease. The next screen (Figure 5.30) prompts the clinician to select one of the test files available to the DDx recommender. Let's say in this scenario, the clinician selects the hypertension file of test data, and clicks the "Proceed >>" button (Figure 5.30). So the next screen (Figure 5.31) asks the clinician if he/she would like to add records to the test records or use the existing test records on file. Let's say in this scenario that the clinician would like to add test records so the clinician selects the "Add Records to the Test Data File then Classify" option and clicks the "Proceed >>" button (Figure 5.31). The following screen allows the clinician to enter a new hypertension patient case. The clinician would be prompted to enter the patient's age, sex, systolic and diastolic blood pressures. The clinician enters the required information (Figure 5.32). Now the clinician can enter another record, and let's say the clinician decides to enter another record, so he presses the "Enter Another Record" button (Figure 5.32). The clinician enters another record and decides not to enter any more records, so she/he clicks the "Proceed >>" button (Figure 5.33). Now the DDx recommender applies learned diagnosis trends to the test data, and makes diagnosis predictions for the test data. It writes the diagnosis predictions in a predictions file. For each test file available to the proximity-based DDx

recommender, there is a corresponding predictions file that contains diagnosis predictions for each test record in the test file. Therefore, the diagnosis predictions file has the same patient records in the test file but with the diagnosis predictions added. The test file has patient cases with no diagnosis indicated. The test file for this scenario is given in figure 5.34, and the corresponding predictions file is given in figure 5.35 for comparison purposes.

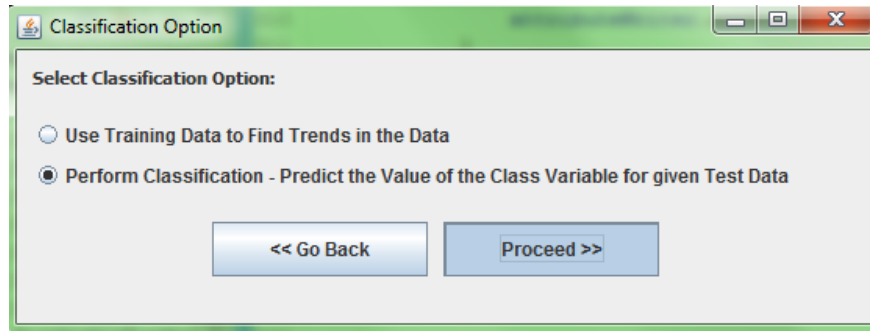


Fig. 5.29: Selecting Second Classification Option for the D1 Relation of the Proximity-based DDx Recommender

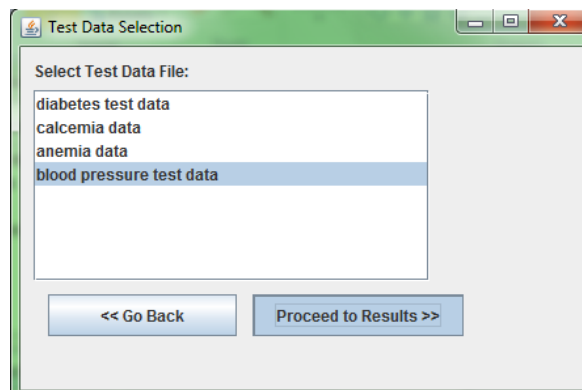


Fig. 5.30: Test Data File Selection Screen of the Proximity-based DDx Recommender

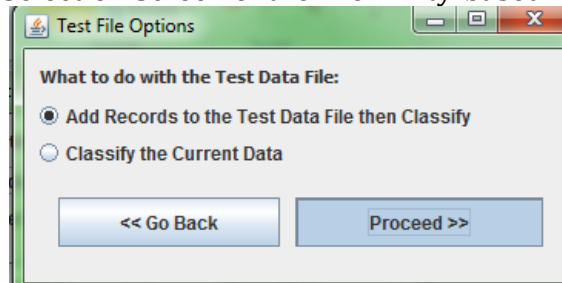


Fig. 5.31: Test Data Records Screen of the Proximity-based DDx Recommender

Fig. 5.32: Screen for Adding New Hypertension Test Record to the Hypertension Test Records File of the Proximity-based DDx Recommender

Fig. 5.33: Adding Another New Hypertension Test Record to the Hypertension Test Records File of the Proximity-based DDx Recommender

50.0,	0.0,	1.0,	60,	49,	?
83.0,	0.0,	1.0,	69,	56,	?
33.0,	0.0,	0.0,	66,	58,	?
71.0,	0.0,	1.0,	156,	101,	?
57.0,	0.0,	1.0,	148,	114,	?
74.0,	0.0,	0.0,	72,	40,	?
62.0,	0.0,	1.0,	142,	103,	?
38.0,	0.0,	0.0,	141,	108,	?
51.0,	0.0,	1.0,	157,	111,	?
81.0,	0.0,	1.0,	142,	99,	?
53.0,	0.0,	1.0,	167,	94,	?
62.0,	0.0,	1.0,	147,	91,	?
61.0,	0.0,	1.0,	157,	95,	?
23.0,	0.0,	1.0,	142,	89,	?
54.0,	0.0,	0.0,	140,	91,	?
45.0,	0.0,	1.0,	128,	67,	?

Fig. 5.34: Hypertension Test File of the Proximity-based DDx Recommender

62.0,	0.0,	0.0,	152,	114,	"HYPERTENSION NOS"
55.0,	0.0,	0.0,	154,	94,	"HYPERTENSION NOS"
26.0,	0.0,	1.0,	87,	48,	"ORTHOSTATIC HYPOTENSION"
57.0,	0.0,	1.0,	165,	108,	"HYPERTENSION NOS"
87.0,	0.0,	1.0,	83,	40,	"ORTHOSTATIC HYPOTENSION"
63.0,	0.0,	0.0,	159,	112,	"HYPERTENSION NOS"
29.0,	0.0,	1.0,	86,	44,	"ORTHOSTATIC HYPOTENSION"
72.0,	0.0,	1.0,	69,	51,	"ORTHOSTATIC HYPOTENSION"
68.0,	0.0,	1.0,	87,	53,	"ORTHOSTATIC HYPOTENSION"
49.0,	0.0,	1.0,	158,	96,	"HYPERTENSION NOS"
57.0,	0.0,	0.0,	60,	55,	"ORTHOSTATIC HYPOTENSION"
67.0,	0.0,	1.0,	160,	103,	"HYPERTENSION NOS"
34.0,	0.0,	1.0,	162,	100,	"HYPERTENSION NOS"
67.0,	0.0,	1.0,	141,	117,	"HYPERTENSION NOS"
47.0,	0.0,	0.0,	66,	55,	"ORTHOSTATIC HYPOTENSION"
53.0,	0.0,	0.0,	164,	111,	"HYPERTENSION NOS"
45.0,	0.0,	1.0,	60,	41,	"ORTHOSTATIC HYPOTENSION"
76.0,	0.0,	1.0,	166,	95,	"HYPERTENSION NOS"
81.0,	0.0,	1.0,	143,	104,	"HYPERTENSION NOS"
76.0,	0.0,	1.0,	150,	97,	"HYPERTENSION NOS"
41.0,	0.0,	1.0,	161,	102,	"HYPERTENSION NOS"
54.0,	0.0,	0.0,	153,	98,	"HYPERTENSION NOS"
41.0,	0.0,	0.0,	157,	109,	"HYPERTENSION NOS"
68.0,	0.0,	0.0,	163,	104,	"HYPERTENSION NOS"
58.0,	0.0,	0.0,	158,	101,	"HYPERTENSION NOS"
87.0,	0.0,	1.0,	151,	109,	"HYPERTENSION NOS"
40.0,	0.0,	1.0,	158,	115,	"HYPERTENSION NOS"
93.0,	0.0,	1.0,	157,	114,	"HYPERTENSION NOS"
40.0,	0.0,	0.0,	164,	96,	"HYPERTENSION NOS"
37.0,	0.0,	1.0,	155,	119,	"HYPERTENSION NOS"
82.0,	0.0,	0.0,	75,	58,	"ORTHOSTATIC HYPOTENSION"
82.0,	0.0,	1.0,	63,	47,	"ORTHOSTATIC HYPOTENSION"
50.0,	0.0,	0.0,	162,	99,	"HYPERTENSION NOS"
77.0,	0.0,	1.0,	88,	41,	"ORTHOSTATIC HYPOTENSION"
69.0,	0.0,	0.0,	153,	96,	"HYPERTENSION NOS"
50.0,	0.0,	1.0,	60,	49,	"ORTHOSTATIC HYPOTENSION"
83.0,	0.0,	1.0,	69,	56,	"ORTHOSTATIC HYPOTENSION"
33.0,	0.0,	0.0,	66,	58,	"ORTHOSTATIC HYPOTENSION"
71.0,	0.0,	1.0,	156,	101,	"HYPERTENSION NOS"
57.0,	0.0,	1.0,	148,	114,	"HYPERTENSION NOS"
74.0,	0.0,	0.0,	72,	40,	"ORTHOSTATIC HYPOTENSION"
62.0,	0.0,	1.0,	142,	103,	"HYPERTENSION NOS"
38.0,	0.0,	0.0,	141,	108,	"HYPERTENSION NOS"
51.0,	0.0,	1.0,	157,	111,	"HYPERTENSION NOS"
81.0,	0.0,	1.0,	142,	99,	"HYPERTENSION NOS"
53.0,	0.0,	1.0,	167,	94,	"HYPERTENSION NOS"
62.0,	0.0,	1.0,	147,	91,	"HYPERTENSION NOS"
61.0,	0.0,	1.0,	157,	95,	"HYPERTENSION NOS"
23.0,	0.0,	1.0,	142,	89,	"Normal"
54.0,	0.0,	0.0,	140,	91,	"HYPERTENSION NOS"
45.0,	0.0,	1.0,	128,	67,	"Normal"

Fig. 5.35: Hypertension Diagnosis Predictions File of the Proximity-based DDX Recommender

First D2 Scenario

In this scenario, the clinician would like to find diagnosis rules relating various clinical patients' attributes and lab test data on one hand, and diagnosis of a certain disease on the other hand. These rules are called association rules. The clinician can access the association rules component from the main window of the proximity-based DDX recommender. The clinician clicks the "Answer D2" button in order to access the association rules component (Figure 5.36). Next, the proximity-based DDX recommender asks the clinician to select the disease diagnosis training data he/she wishes to find association rules for. The clinician selects one of the five available disease diagnosis training data related to the diagnosis of Diabetes, Anemia, Calcemia, and Hypertension. Let's say in this scenario the clinician selects the Anemia data file (Figure 5.37). Each record in the Anemia training data file contains a patient's age and sex, a percentage of red blood cells in a blood sample of the patient's blood, hemoglobin levels in the blood sample, and a diagnosis of positive for anemia or negative (Normal). The association rules algorithm studies these data records to find rules relating any one or a group of the clinical attributes (in this case age, sex, hemoglobin levels, and percentage of red blood cells are the clinical attributes) to the diagnosis result. These rules have confidence factors determined by the percentage of the

training data records they apply to. The rules with high confidence factors are the most interesting. The resulting association rules from this case are presented in **Appendix C**.

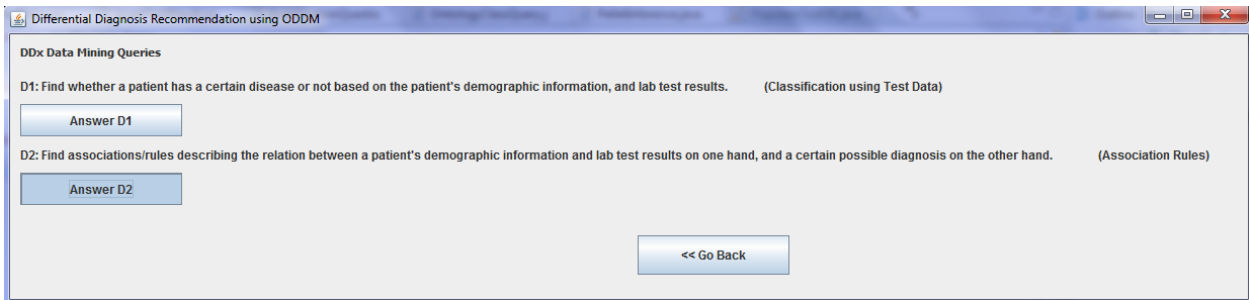


Fig. 5.36: Accessing the Association Rules Component of the Proximity-based DDX Recommender

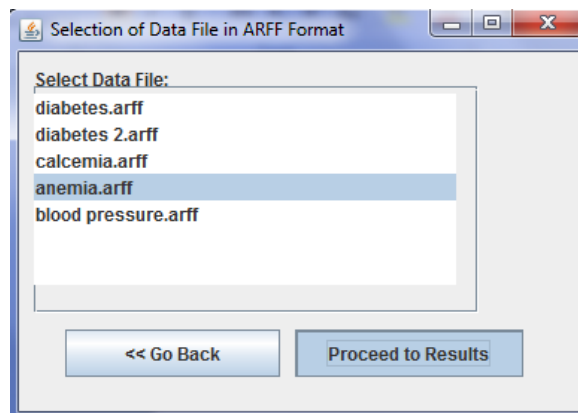


Fig 5.37: Selection of the Diagnosis Data for the Apriori Association Rules Algorithm of the Proximity-based DDX Recommender

Chapter 6: Conclusions and Future Research

6.1. Conclusions

We presented in this thesis a new Differential Diagnosis (DDx) recommendation model. It is a comprehensive model for computer-aided disease diagnosis recommendation. The realization of this comprehensive model requires a number of resources to be researched and developed first. In this thesis, we identified the following resources to be lacking or currently unavailable:

Suitable medical diagnosis ontology: The amount of information that must be taken into account in medical diagnosis is huge and subject to evolution. Ontologies are a means for formalizing the concepts of the domain of interest. In our model, one important function of the DDx recommender is to be able to make connections between symptoms and diseases. Unfortunately, there is no ontology that makes relations between symptoms and diseases. In chapter 2, we developed a “kernel” ontology that makes such connection. We call this ontology the Diseases Symptoms Ontology (DSO). The development of the DSO is based on aligning two already existed standard/OBO ontologies (DOID and SYMP). The two OBO ontologies provide essential knowledge representation for diseases and symptoms as a separate model, but not the relations between them (e.g. what are the symptoms of each disease). As these relations are required in the reasoning process of our DDx recommendation model, we have created kernel diseases symptoms ontology (DSO) combining DOID & SYMP and thus combining diseases and symptoms into one ontology structure. Although the developed kernel DSO defines the causality relations between diseases and symptoms for a dozen diseases and their related symptoms, the DSO structure and the alignment techniques used allow the expansion to create further relationships between additional diseases and their related symptoms. However, since the numbers of diseases are quite huge (e.g. DOID identified over than 8000 diseases), and since for each disease there are a number of related symptoms, a full implementation of the DSO would require the creation of tens of thousands of causality relations between diseases and symptoms so that is a large amount of work. Therefore, we believe that the development of a complete DSO ontology is an open community-wide research challenge. In chapter 2, we introduced a systematic way to allow a research community effort and to our future research to efficiently accomplish creating a more complete DSO.

Suitable Patient Ontology: The proliferation of medical terms related to patient disease diagnosis is a major obstacle in the sharing of medical information (e.g. electronic medical records, emergency forms, and triage notes) among shareholders (e.g., hospitals, clinicians, pharmaceutical companies etc.). Hence there is a need for a uniform structure that contains attributes from various medical documents and relations between these attributes. Chapter 4

addresses this problem and introduces a patient ontology (PO). The developed patient ontology contains attributes which are commonly found in various patient medical records. In our prototype, the PO is used to assist the DDx recommender in selecting attributes that are relevant to a certain diagnosis.

Suitable Clinical Pathways Rules/Ontology: Since 2006 there were many attempts to start developing adaptive clinical pathways for disease diagnosis and treatment. For example, Helen Chen [Chen 2006] of Agfa Healthcare proposed to initiate a W3C AHPP taskforce within the W3C Semantic Web for Health Care and Life Sciences Interest Group (<http://www.w3.org/2001/sw/hcls/>). Clinical pathways are becoming standardized tools to translate evidence-based recommendations into locally practicable, process-specific algorithms that reduce practice variations and optimize quality of care. However, clinical pathways are only known in a form that is only readable to humans and not machines (e.g. care maps⁷¹). Chapter 4 described the many attempts to establish a machine readable clinical pathways ontologies that ended with no real success as the problem of creating a comprehensive clinical pathways acceptable to all healthcare institutions for large number of diseases represent a huge task that requires healthcare community-wide efforts. Given the lack of such a comprehensive clinical pathways ontology, we developed in chapter 4 an alternative method to create adaptive clinical pathways in a form of dynamic rules. However, creating a clinical pathway rules for even one disease could require hundreds of interrelated rules. Actually the task of creating clinical pathways rules for all the known diseases needs the creation of hundreds of thousands of rules. Chapter 4 described a simplified method for creating these dynamic rules which can be placed in an external text file (Drools File in our prototype) and using a format that can be easily understood by clinicians for the purpose of adding new rules or modifying existing rules. In our prototype we used the described method and created the clinical pathways rules for the diagnosis and management of four diseases. This method can be repeated for any number of other diseases.

Suitable Reference Medical Diagnosis Medical Dataset: The third lacking resource is patient diagnosis data. Much of the diagnosis data ranging from a patient's demographic information (age, weight, height, sex, etc) to lab test results are not publicly available. Therefore, we found it difficult to obtain such data. For some of the more common diseases like diabetes and hypertension data for some of the diagnosis variables are publicly available. We have used this data in our prototype specifically we used the NIS dataset (see chapter 4). The lack of the availability of diagnosis data limits the choices, in front of the proximity-based recommender, of diagnosis variables to those variables where data values are available. This negatively affects the work the DSO & PO ontologies do for the proximity-based recommender. As mentioned in chapter 4, the DSO & PO ontologies select the relevant diagnosis variables for a specified diagnostic case. The proximity-based recommender is supposed to take these diagnosis variables,

⁷¹ http://en.wikipedia.org/wiki/Care_map

form a table called the training table made up of values for these diagnosis variables and the diagnosis prediction obtained from patient records, and this data table to data mining algorithms so they can predict diagnosis for new cases for the particular diagnosis of interest as well as generate diagnosis rules for the diagnosing the specific diagnosis. Data mining predictions depend on the quality of the available data. For the selected diagnosis variables for which data is not available or lacking, these variables cannot be used by the proximity-based recommender in the data mining process that results in diagnosis predictions. The DSO & PO ontologies determined that a set of variables should be considered for a certain diagnosis, yet the proximity-based recommender cannot consider any of the variables unless it has sufficient data for a variable and this data must also be related to the specific diagnostic case. Therefore, the lack of data would render required diagnostic variables unusable and this is a hindering factor for a DDX recommendation process which relies on data mining.

Besides creating the essential resources for building effective medical diagnosis systems, our DDX recommendation model identifies the following novel components and methods to enable disease diagnosis prediction:

1. DDX Evidence-based Relational Query Engine (DSO Ontology Crawler)

The **ontology crawler** is the component primarily used in our differential diagnosis engine for retrieving information from the DSO ontology using representative querying relations that are likely to be asked by clinicians when trying to diagnose diseases. Our DSO ontology crawler uses six representative relations (R1-R6 see chapter 3) to address the common queries that health care professionals use during the process of differential diagnosis. Figure 3.3 that was introduced in Chapter 3 lists these relations along with their formal representation. Our DSO ontology crawler component consists of the Apache Jena semantic web framework⁷², which we used to our program the six relations for combining the ontological attributes in a meaningful way for the purpose of inferring new knowledge necessary for the differential diagnosis process. The most significant part of this component is that our ontology crawler replaces the SPARQL primitive querying engine by implementing the necessary clinical differential diagnosis relations using Jena OWL API primitives. Our ontology crawler has been designed to act as a kernel component of our DDX recommendation model and hence it is an important part of the overall architecture that aims to serve health care providers with the notion of a differential diagnosis recommendation system. We find it useful to re-introduce Figure 3.3 from chapter 3 briefly illustrates the six differential diagnosis querying relationships.

2. Ontology-Driven Evidence-based DDX Recommendation

The second important component in our DDX recommendation model is the ontology-driven evidence-based recommendation component. This component is built on our ontology crawler to

⁷² <http://jena.sourceforge.net/>

represent the standardized rules and procedures used for diagnosing diseases. In this direction the component identified twelve common diseases and implemented their clinical pathways rules.

The transformation of clinical pathway knowledge and other medical knowledge for disease diagnosis into rules is a prerequisite for our evidence-based DDx recommender. The evidence-based DDx recommender employs these rules in a forward chaining process driven by the Drools rule engine to find appropriate diagnosis recommendations. The rule firing process is initiated by the DSO & PO ontologies (hence ontology-driven). The DSO & PO ontologies select the clinical data most relevant to a specific diagnostic case and feed it to the rule engine. Upon receiving the selected data, the rule engine initiates the forward chaining rule firing process. It compares the data to the clinical pathway rules in order to reach diagnostic conclusions. The ontologies optimize the rule firing process by refining the data input to only include data relevant to a specific diagnostic case. The evidence-based component utilizes relational database components (we used Apache Derby⁷³ database and Hibernate DB driver⁷⁴) to store patient data as well as an ontology reasoner (we used Pellet⁷⁵). Figure 4.3 of Chapter 4 is an illustration of the ontology-driven evidence-based DDx recommendation model. For more information on the evidence-based DDx recommender, please see chapter 4.

3. Ontology-Driven Proximity-based DDx Recommendation

This is the third major component of our DDx recommendation model, which is built upon the ontology crawler and the evidence-based recommendation component. The sole function of this component is to enable clinicians to predict the diagnosis of certain new cases based on some previously available training data. Because of the predictive nature of this component's functionality, we call it the proximity-based recommendation component. This component has two major tasks. The first is to extract, from patient data records, the relevant symptoms and test data for a certain diagnostic case based on semantic web technologies for matching and navigation, with the aid of the DSO and the PO ontologies. The matching and navigation process is made on the data produced by the evidence-based component or any training data provided by other clinical systems or medical repositories. Through the PO and DSO ontologies and their crawlers (query engines), this component will select relevant patient attributes, lab test attributes, and certain possible diseases under consideration for diagnosis. Training data for the selected clinical variables will be then used by the second task that involves data mining classification algorithms to learn diagnosis trends. After that training, these classification algorithms use the learned diagnosis trends to predict the diagnosis for provided test data (i.e. new clinical data cases that require diagnosis). Note that learned diagnosis trends are only valid predictors of diagnosis for test data, when the clinical attributes of the test data largely match the clinical attributes of the training data. The second task may use an alternative group of data mining

⁷³ <http://db.apache.org/derby/>

⁷⁴ <http://www.hibernate.org/>

⁷⁵ <http://clarkparsia.com/pellet/>

algorithms that involves applying association algorithms on the training data to learn rules relating diagnosis variables (clinical attributes) with other diagnosis variables, and rules relating diagnosis variables and possible diagnoses. These newly generated rules can be used to make diagnosis predictions/recommendations. We programmed this component using an incorporation of the Weka Java API along with other predictive and analytical components. Figure 4.5 from chapter 4 is a good illustration of the proximity-based DDx recommender otherwise called the ontology-driven data mining DDx recommender. For more information on this component, please see chapter 4.

4. DDx Integration (Evidence-based with Proximity-based Diagnosis)

This is our fourth major component where we integrate the first three components to form our overall DDx recommendation model. The evidence-based and proximity-based approaches for DDx recommendation can cooperate to form an overall DDx recommendation model. Results generated by the evidence-based DDx recommender can be used for prediction by the data mining algorithms of the proximity-based DDx recommender. On the other hand, data mining algorithms from the proximity-based approach generate rules that can be used to update the rule base of the evidence-based approach. Figure 4.1 re-introduced from Chapter 4 illustrate our overall DDx integral recommendation model. Figure 3.2 re-introduced from chapter 3 illustrates the various API's used in our prototype implementation of our overall DDx integral recommendation model.

6.2. Future Research

Here is a list of some of the possible research extensions and expansions from our current research work:

1. Extending the DSO Ontology to cover related diseases groups (e.g. Geriatric DSO that links all the common diseases and their symptoms for people of old age). This research extension is quite important as it will support the new initiative of medical diagnosis knowledge personalization [Cristina Romero-Tris, David Riaño and Francis Real 2010]. The extended DSO need to incorporate the ICD-10 coding for the related disease groups. It is important to note that the current DSO Ontology includes twelve common diseases that share some common symptoms and signs.
2. Enhancing the PO Ontology to conform to the requirements of major EMR (Electronic Medical Record) standards like HL7 (<http://www.hl7.org/ehr/>). The current PO has been developed to be rather a generic ontology that can work with any EMR standard. However, conforming to HL7 standard will enable our DDx recommendation system to build an HL7 messaging interface to enables physicians and healthcare institutions to connect with each other. We propose in this direction to use the InterSystems Ensemble APIs

(<http://www.intersystems.com/hl7/index.html>) to develop such messaging interface that can be added to our current DDx recommendation system.

3. Although the use of Drools rules⁷⁶ provides a convenient and intuitive way for clinicians to describe and classify diagnostic knowledge contents for our DDx recommendation system, it remains hard to use these rules with other automated software agents for reasoning. In this direction we are proposing to convert the rules that are related to a specific disease or group of related diseases into OWL ontologies, thus enabling reasoning on them by other semantic web applications. This conversion process will assist in the process of producing *incremental clinical pathways ontologies (ICPO)* for specific diseases or for a group of related diseases. We are proposing to perform the conversion of rules into OWL ontology using the method introduced by Fausto Giunchiglia, Ilya Zaihrayeu, and Feroz Farazi [Giunchiglia *et al.* 2008]. The process of producing ICPO needs to use a sound ontology enrichment technique. Ontology enrichment is the activity of extending ontology by adding new elements (e.g. concepts, relations, properties, axioms) [Castano 2007].
4. Expanding the integration between the evidence-based and the proximity-based diagnosis components. Our current integration concentrates much on linking the evidence-based component to the proximity-based by adding the confirmed diagnosis from the evidence-based component to the training dataset of the proximity-based component. In our integration model, we have proposed that diagnosis rules produced by the proximity-based component be added to, or override, or confirm, or negate existing diagnosis rules in the rule set of the evidence-based component. However, the rules produced by the proximity-based component are in a different format from the rules of the evidence-based component. Therefore, the rules from the proximity-based component must be converted into the rule format of the evidence-based component in order for proper comparison of the rules to be possible.
5. Expanding our DDx recommender querying relations. In this direction we are intending to generate the second level of DDx querying relations so clinicians can specifically query the DDx recommender about a specific disease or a group of related diseases. Our six DDx relations developed in this thesis represent the most generic questions used by the clinicians to diagnose any unknown disease.
6. Further comparing the different classification and association data mining algorithms and their efficiency in accurately predicting disease diagnosis. We are intending to use the set of confirmed diagnosis cases as produced by the evidence-based component and hide their final diagnosis for analysing the different data mining diagnosis accuracy prediction. We are also proposing to experiment with other prediction techniques like clustering in predicting diagnosis for group of related diseases.

⁷⁶ <http://www.jboss.org/drools>

7. There are many possible implementations for our DDx recommendation model into useful healthcare applications. The following are some of the proposed direct applications:
- a. **Application 1** - Home Care DDx Recommender: This application requires the addition of mobile client interfaces (e.g. Android or iPhone) to monitor some of the patient's parameters related to certain chronic diseases like diabetes and hypertension. We are proposing to use the MORF technology [Benlamri and Dockstader 2010].
 - b. **Application 2** - Web Clinicians Disease Diagnosis Recommender: This application will enable clinicians to use our DDx recommender from the web. This could be a web implementation using the evidence-based component and/or the proximity-based component.
 - c. **Application 3** - Web Discharge and Referral System: This application will enable clinicians to build a patient disease diagnosis record for discharge or referral purposes. We are proposing to package the discharge summary in one of the most popular formats like the Continuity of Care records⁷⁷ or the Continuity of Care Document⁷⁸.
 - d. **Application 4** - Collaborative DDx Recommender. In this application we are proposing to extend our current DDx recommender prototype to add a social networking environment for a clinical community of practice. For this purpose we are proposing to integrate one of the open-source social networking engines like Elgg⁷⁹ with our DDx recommendation architecture. This extension will enable the healthcare community of practice to collaboratively use our DDx recommender for more effectively diagnosing diseases as well as for community collaboration and learning.
 - e. **Application 5** – DSL Based DDx Recommender: In this application we are proposing to replace the various UI components by domain specific natural language (DSL) commands. In this direction we need to define the required syntax of the clinician's DSL commands and interactions as well as to identify their parsing and processing components.

⁷⁷ <http://www.ccrstandard.com/>

⁷⁸ http://en.wikipedia.org/wiki/Continuity_of_Care_Document

⁷⁹ <http://elgg.org>

Research Publications

Mohammed, O., Benlamri, R.; *Building a Diseases Symptoms Ontology for Medical Diagnosis: An Integrative Approach*, To be published (2012)

Mohammed, O., Benlamri, R.; *Building an Semantic Web Model for Differential Diagnosis Recommendation*, To be published (2012)

References

- Abidi, S. (2008), *Healthcare Knowledge Management: the Art of the Possible*, In: K4CARE 2007, LNCS (LNAI), vol. 4924, pp. 1–20, Riaño, D. (ed.), Springer-Verlage, Berlin, Germany
- Adlassnig, K.P. (1980), *A fuzzy logical model of computer assisted Medical Diagnosis Method*, Info Med., vol. 19, pp.141-148
- Agrawal, R.; Ramakrishnan, S. (1994), *Fast algorithms for mining association rules in large databases*, In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pp. 487-499., Santiago, Chile, September 1994, Available Online: <http://rakesh.agrawal-family.com/papers/vldb94apriori.pdf>
- Akay, M. (2008), *Biomedical engineering for global healthcare*, 8th IEEE International Conference on BioInformatics and BioEngineering, BIBE 2008, Athens, Greece, 8-10 Oct. 2008
- Alexandrou, D.; Xenikoudakis, F.; Mentzas, G. (2009), *SEMPATH: Semantic Adaptive and Personalized Clinical Pathways*, eTELEMED '09, International Conference on eHealth, Telemedicine, and Social Medicine, 2009, 1-7 Feb. 2009, Cancun, pp. 36 – 41
- Al-Shayea, Qeethara Kadhim (2011), *Artificial Neural Networks in Medical Diagnosis*, IJCSI International Journal of Computer Science Issues, vol. 8, Issue 2, March 2011, ISSN (Online): 1694-0814 www.IJCSI.org
- Al-Qaysi, I. *et al.* (2010), *Holonic and Optimal Medical Decision Making Under Uncertainty*, In: 2010 IEEE EMBS Conference on Biomedical Engineering & Sciences (IECBES 2010)-Rehabilitation Engineering & Technology, Kuala Lumpur, Malaysia, 30 November to 2nd December 2010
- Anderson, G., Horvath, J. (2004), *The Growing Burden of Chronic Disease in America*, Public Health Reports, vol. 119, issue 3, pp. 263–270, Available Online: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1497638/pdf/15158105.pdf>
- Anderson, E. R. (2007), *Fuzzy and Rough Techniques in Medical Diagnosis and Medication*, Springer Verlage
- Ángel García-Crespo, *et al.* (2010), *ODDIN: Ontology-driven differential diagnosis based on logical inference and probabilistic refinements*, Journal of Expert Systems with Applications vol. 37, issue 3, pp. 2621-2628, 15 March 2010
- Audimoolam, S.; Nair, M.; Gaikwad, R.; Qing, C. (2005), *The Role of Clinical Pathways in Improving Patient Outcomes*, Graduate Student Technical Report, 7th February, Available Online: <http://www.cs.dal.ca/~sraza/StudentWork/clinicalpathwayspaper.pdf><http://web.cs.dal.ca/~asharma/OWL/Role%2520of%2520Clinical%2520Pathways.pdf>
- Balcázar, J.L. (2009), *Confidence Width: An Objective Measure for Association Rule Novelty*, Workshop on Quality issues, measures of interestingness and evaluation of data mining models QIMIE'09 at PAKDD'09

- Barnett, G.O.; Cimino, J.J.; Hupp, J.A.; Hoffer, E.P. (1987), *An evolving diagnostic decision-support system*, In: JAMA The Journal Of The American Medical Association, vol. 258, issue 1, pp. 67-74
- Benlamri, Rachid; Dockstader, Luke (2010), *MORF: A Mobile Health-Monitoring Platform*, IT Professional, vol. 12, issue 3, pp. 18-25
- Berner, E.S. (ed.) (2007), *Clinical Decision Support Systems*, Springer New York, NY, USA
- Billie, Erson; Davis, C.M. ; Hardin, J.M. (2010), *Chapter 12: Using Data Mining to Build Alerting Systems for Decision Support in Healthcare*, In: Healthcare Informatics: Improving Efficiency and Productivity, Kudyba, S.P. (ed.), CRC Press ISBN: 978-1-4398-0978-5
- Bloehdorn, S.; Haase, P.; Huang, Z.; Sure, Y.; Völker, J.; van Harmelen, F.; Studer, R. (2009), *Ontology Management*, Davies, J.; Grobelnik, M.; Mladenic, D. (eds.), Semantic Knowledge Management, Springer-Verlage, pp. 3–20
- Bloehdorn, S.; Haase, P.; Sure, Y.; Voelker, J. (2009), *Ontology Evolution*, pp. 51-70, John Wiley & Sons
- Bodenheimer, T.; Wagner, E.H.; Grumbach, K. (2002), *Improving Primary Care for Patients with Chronic Illness*, The Journal of the American Medical Association, vol. 288, issue 14, pp. 1775–1779
- Bouamrane, M.M.; Retor, Alan; Hurrell, Martin (2010), *Experience of Using OWL Ontologies for Automated Inference of Routine Pre-operative Screening Tests*, 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, The Semantic Web, Revised Selected Papers part 2, Lecture Notes for Computer Science (LNCS), Schneider, Peter F.P. et al. (ed.)
- Bouckaert, Remco R.; Frank, Eibe; Hall, Mark A.; Holmes, Geoffrey; Pfahringer, Bernhard; Reutemann, Peter; Witten, Ian H. (2010), *WEKA-experiences with a java open-source project*, *Journal of Machine Learning Research*, vol. 11, pp. 2533-2541
- Brisson, Laurent; Collard, Martine (2008), *AN ONTOLOGY DRIVEN DATA MINING PROCESS*, In Proceedings of ICEIS (2)'2008, Barcelona, Spain, pp.54-61,
<http://rainbow.polytech.unice.fr/publis/brisson-collard:2009.pdf>
- Braun, Simone; Schmidt, Andreas; Walter, Andreas; Nagypal, Gabor; Zacharias, Valentin (2007), *Ontology Maturing: a Collaborative Web 2.0 Approach to Ontology Engineering*, Noy, Natasha; Alani, Harith; Stumme, Gerd; Mika, Peter; Sure, York; Vrandecic, Denny (eds.), Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge (CKC 2007) at the 16th International World Wide Web Conference (WWW2007), CEUR Workshop Proceedings vol. 273, Banff, Canada, May 8, 2007, Available Online:
http://www2007.org/workshops/paper_14.pdf
- Brown, Lance, et al. (2002), *The Critically Ill Or Comatose Infant: An Organized Approach*, Journal of Emergency Medicine Practise, vol. 4, issue 10, October 2002
- Buranarach, Marut, et al. (2009), *A Semantic Web Framework to Support Knowledge Management in Chronic Disease Healthcare*, Available Online: http://text.hlt.nectec.or.th/marut/papers/dcare_mtrs09.pdf

Castano, S., et. al (2007), *Ontology Dynamics with Multimedia Information: The BOEMIE Evolution Methodology*, In: Proc. of International Workshop on Ontology Dynamics (IWOD) ESWC 2007 Workshop, Innsbruck, Austria, June 7, 2007, Available Online: <http://kmi.open.ac.uk/events/iwodi/papers/paper-07.pdf>

Catley, C.; Frize, M.; Walker, C.R.; St. Germain, L. (2003), *Integrating Clinical Alerts into an XML-Based Health Care framework for the Neonatal Intensive Care Unit*, In: 25th IEEE EMBS, Cancun, Mexico

Chabalier, Julie; Dameron, Olivier; Burgun, Anita (2007), *Integrating and querying disease and pathway ontologies: building an OWL model and using RDFS queries*, The 10th Annual Bio-Ontologies Meeting, Co-Located with ISMB/ECCB 2007, Austria, July 2007, Available Online: http://www.ea3888.univ-rennes1.fr/lim/doc_184.pdf

Chen, Helen; Colaert, Dirk; De Roo, Jos (2004), *Towards Adaptable Clinical Pathway Using Semantic Web Technology*, Position Paper, July 2004, W3C Workshop Semantic Web for Life Science, Available Online: <http://www.w3.org/2004/07/swls-cfp.html>

Chen, Helen; Abidi, S.R. R. (2006), *Adaptable personalized care planning via a semantic web framework*, 20th International Congress of the European Federation for Medical Informatics (MIE 2006), Maastricht, Netherlands, August 27-30, 2006

Chen, Helen (2006), *Semantic Web in Adaptable Health Care Protocols and Pathways Group Charter - A proposal*, February 2006

Chien-Chih, Wang; Ming-Nan, Chien; Chua-Huang, Huang; Liu, Li (2007), *A Rule-Based Disease Diagnostic System Using a Temporal Relationship Model*, Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007, pp. 109 - 115, Haikou, China, 24-27 Aug. 2007

Colucciello, Stephen A.; Lukens, Thomas W.; Morgan, David L. (1999), *Assessing Abdominal Pain In Adults: A Rational, Cost-Effective, And Evidence-Based Strategy*, Journal of EMERGENCY MEDICINE PRACTICE, vol. 1, issue 1

Cohen, W.W. (1995), *Fast Effective Rule Induction*, In: Twelfth International Conference on Machine Learning (ICML), pp.115-123, Tahoe City, California, USA, July 9-12, 1995

Concaro, Stefano *et al.*, *Temporal data mining methods for the analysis of the AHRQ archives*, Università di Pavia, Pavia , Italy

Cowell, Lindsay; Smith, Barry (2010), *Infectious Disease Ontology*, In Vitali Sintchenko, Infectious Disease Informatics, Springer Verlage

Cray, Marc Imhotep (2011), *Subjects and Topics in Medical Science: An Imhotep Virtual Medical School Primer*, Available Online: <http://www.scribd.com/doc/62730285/34/Differential-diagnosis>

De Luc, Kathryn; Todd, Julian (eds.) (2003), *e-Pathways: computers and the patient's journey through care*, THE NEW ZEALAND MEDICAL JOURNAL, vol. 116, issue 1184, pp. 1-2, 24 October 2003

De Nicola, Antonio; Missikoff, Michele; Navigli, Roberto (2009), *A Software Engineering Approach to Ontology Building*, Information Systems (Elsevier), vol. 34, issue 2, pp. 258–275, Available Online: http://www.dsi.uniroma1.it/~navigli/pubs/De_Nicola_Missikoff_Navigli_2009.pdf

Denney, Christine, *et al.* (2009), *Creating a Translational Medicine Ontology*, W3C Proceedings, Available online: <http://proceedings.nature.com/documents/3686/version/1/files/npre20093686-1.pdf>

Diamond, H. (2004), *Clinical Reminder System: A Relational Database Application for Evidence-Based Medicine Practice*

Doan, AnHai, *et al.* (2003), *Learning to match ontologies on the Semantic Web*, The VLDB Journal — The International Journal on Very Large Data Bases archive, vol. 12, issue 4, November 2003

Domingos, P.; Pazzani, M. (1997), *On the Optimality of the Simple Bayesian Classifier under Zero-One Loss*, In: Journal of Machine Learning, vol. 29, issue 2 - 3, pp. 103-130

Dumontier, Michel (2010), *Building an effective Semantic Web for Health Care and the Life Sciences*, Semantic Web Journal (SWJ), vol. 1, issue 1-2, pp. 131-135, IOS Press, December, 2010, Available Online: <http://iospress.metapress.com/content/m73515426803335m/fulltext.pdf>

Every, N.R.; Hochman, J.; Becker, R.; Kopecky, S.; Cannon, C.P. (2000), *Critical pathways. a review.*, Circulation, 101, pp. 461–465

Famili, A.; Ouyang, J. (2003), *Data Mining: Understanding Data and Disease Modeling*, Proceedings of IASTED-AI-03 Conference, Innsbruck, Austria, February 10-13, 2003

Fernandez-Llatas, Carlos; Meneu, Teresa; Bened'1, Jose Miguel; Traver, Vicente (2010), *Activity-Based Process Mining for Clinical Pathways Computer Aided Design*, 32nd Annual International Conference of the IEEE EMBS, Buenos Aires, Argentina, August 31 - September 4, 2010, Available Online: http://heartcycle.med.auth.gr/resources/publications/activity_based_mining.pdf

Gamberger, Dragan; Lavrac, Nada; Jovanoski, Viktor (1999), *High confidence association rules for medical diagnosis*, Proceeding of Intelligent data analysis in medicine and pharmacology (IDAMAP'99), pp. 42-51, Washington, D.C., USA

Giunchiglia, Fausto; Zaihrayeu, Ilya; Farazi, Feroz (2008), *Converting Classifications into OWL Ontologies*, Technical Report DISI-08-027, Ingegneria e Scienza dell'Informazione, University of Trento, Available Online: <http://eprints.biblio.unitn.it/archive/00001439/>

Goodnough *et al.* (2005), *Detection, Evaluation, and Management of Anemia in the Elective Surgical Patient*, the International Anesthesia Research Society

Gorunescu, Florin (2007), *Data Mining Techniques in Computer-Aided Diagnosis: Non-Invasive Cancer Detection*, World Academy of Science, Engineering and Technology 34, Available Online: <http://www.waset.org/journals/waset/v34/v34-49.pdf>

Hadzic, Maja; Chang, Elizabeth (2005), *Ontology-based Multi-agent Systems Support Human Disease Study and Control*, Proceeding of the 2005 conference on Self-Organization and Autonomic Informatics, Czap, Hans, Unland, Rainer, Branki, Cherif, Tianfield, Huaglory, IOS Press, vol. 135, pp. 129-141, Glasgow, UK, Dec. 11, 2005

Hamm, Russell; Knoop, S.E.; Schwarz, P.; Block, A.D.; Davis, W.L. (2007), *Harmonizing clinical terminologies: driving interoperability in healthcare*, Loinc Stud Health Technol Inform Report No. 129:660-3, Available Online: <http://loinc.org/articles/Hamm2007a>

Hasan, M.A.; Sher-E-Alam, K.M.; Chowdhury, A.R. (2010), *Human Disease Diagnosis Using a Fuzzy Expert System*, In: JOURNAL OF COMPUTING, vol. 2, issue 6, JUNE 2010, <http://arxiv.org/ftp/arxiv/papers/1006/1006.4544.pdf>

Hepp, M.; Leenheer, P. de; Moor, A. de; Sure, Y. (eds.) (2008), *Ontology Management: Semantic Web, Semantic Web Services, and Business Applications*, Springer series on Semantic Web and Beyond, vol. 7

Hu, Z.; Li, J.S.; Zhou, T.S.; Yu, H.Y.; Suzuki, M.; Araki, K., *Ontology-Based Clinical Pathways with Semantic Rules*, J Med Syst, Springer, March (2011)

Huang, Z.; Lu, X.; Duan, H. (2011), *Using Recommendation to Support Adaptive Clinical Pathways*, J Med Syst., 2011 Jan 5

Hughes, Todd C.; Ashpole, Benjamin C. (2004), *The Semantics of Ontology Alignment*, Presented at I3CO Information Interpretation and Integration Conference, Available Online: <http://www.atl.lmco.com/projects/ontology/papers/SOA.pdf>

Homola, M.; Serafini, L. (2010), *Towards Formal Comparison of Ontology Linking, Mapping and Importing*, Proceeding of 23rd Int. Workshop on Description Logics (DL2010), CEUR-WS 573, Waterloo, Canada, Available Online: http://www.cs.uwaterloo.ca/conferences/dl2010/papers/paper_26.pdf

Isam, M.; Wu, Q.; Ahmadi, M; Ahmad, M. (2007), *Investigating the Performance of NaïveBayes Classifiers and K-NNC*, International Conference on Convergence Information Technology, 2007, pp. 1541-1546

Kappen, Bert *et al.* (2003), *Promedas: A clinical diagnostic decision support system*, Proceedings of the 15th Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2003), pp. 23-24, Netherlands, Available Online: ftp://download.intel.com/research/share/UAI03_workshop/Kappen/Kappen_uai2003.pdf

Kapoor, B.; Sharma, S. (2010), *A Comparative Study Ontology Building Tools for Semantic Web Applications*, International Journal of Web & Semantic Technology (IJWesT), vol. 1, issue 3, pp. 1-13, July, 2010

- Khondoker, M.R.; Mueller, P. (2010), Comparing Ontology Development Tools Based on an Online Survey, In Proceedings of the World Congress on Engineering (WCE), London, UK, June 30 - July 2, 2010, Available Online: http://protege.stanford.edu/publications/Khondoker_Mueller_ComparingOntologyDevelopmentTools.pdf
- Klein, M. (2004), *Change Management for Distributed Ontologies*, PhD thesis, Vrije Universiteit in Amsterdam
- Kohavi, R. (1996), *Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid*, In: Second International Conference on Knowledge Discovery and Data Mining (KDD), vol. 7, pp. 202-207. Portland, Oregon, USA, August 2-4, 1996, Available Online: <http://www.aaai.org/Papers/KDD/1996/KDD96-033.pdf>
- Köhler S. *et al.* (2009), *Clinical diagnostics in human genetics with semantic similarity searches in ontologies*, Am J Hum Genet, vol. 85, issue 4, pp. 457-64
- Kuhn, K.A. (2007), *Medinfo 2007*, vol. 1, IOS Press, Netherlands
- Kumar, A.K.; Singh, Y.; Sanyal S. (2009), *Hybrid approach using case-based reasoning and rule-based reasoning for domain independent clinical decision support in ICU*, Expert Syst. Appl. Int. J., vol. 36, pp. 65-71
- Lin, R.H (2009), *An intelligent model for liver disease diagnosis*, Artificial Intelligence in Medicine Journal, vol. 47, issue 1, pp. 53-62
- Lin, Y. C. (2009), *Development of an Ontology-based Flexible Clinical Pathway System*, WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS, vol. 6, issue 12, December 2009, Available Online: <http://www.wseas.us/e-library/transactions/information/2009/32-900.pdf>
- Liu, Rey-Long; Tung, Shu-Yu; Lu, Yun-Ling (2011), *Identifying Disease Diagnosis Factors by Proximity-Based Mining of Medical Texts*, Intelligent Information and Database Systems, Springer Verlage Lecture Notes in Computer Science, vol. 6592, pp. 171-180
- Luciano, J.S., *et al.* (2011), *The Translational Medicine Ontology and Knowledge Base: Driving personalized medicine by bridging the gap between bench and bedside*, published in J Biomed Semantics, vol. 2, issue 2, pp. S1., May 17, 2011, Available Online: <http://www.w3.org/wiki/images/0/07/Tmo-r4.pdf>
- Martin, F. (2004), *Medical Diagnosis: Test First, Talk Later?*, Mathemedics, Inc., USA
- Marques, Davis (2009), *A Survey of Recent Research in Ontology Mapping*, Simon Fraser University, Technical Report 2009, Available Online: <http://www.sfu.ca/~mhatala/iat881/2005/DM-OntologyMapping.pdf>
- Maniraj, V.; Sivakumar, R. (2010), *Ontology Languages – A Review*, International Journal of Computer Theory and Engineering, vol. 2, issue 6, December, 2010, Available Online: <http://www.ijcte.org/papers/257-G750.pdf>

- Matsumoto, T.; Shimada, Y.; Kawaji, S. (2004), *Clinical diagnosis support system based on symptoms and remarks by neural networks*, 2004 IEEE Conference on Cybernetics and Intelligent Systems, pp. 1304 - 1307, Singapore
- Miller, R.A.; Pople, H.E.; Myers, J.D. (1982), *INTERNIST-1, an experimental computer-based diagnostic consultant for general internal medicine*, In: N Engl J Med 1982, vol. 307, pp. 468-476., <http://www.amia.org/staff/eshortliffe/Clancey-Shortliffe-1984/Ch8.pdf>
- Moein, S.; Monadjemi, S.A.; Moallem, P. (2008), *A Novel Fuzzy-Neural Based Medical Diagnosis System*, In: World Academy of Science, Engineering and Technology, vol. 37, Available Online: <http://www.waset.org/journals/waset/v37/v37-26.pdf>
- National Institute for Health and Clinical Excellence (NHS), *Hypertension: management of hypertension in adults in primary care*, August 2004
- Nikoskelainen, Eeva (2005), *Clinical Pathways in Neuro-Ophthalmology: An Evidence-Based Approach*, Acta Ophthalmologica, vol. 83, Issue 1, Feb 2005
- Northwestern University (2011), Available Online: http://obofoundry.org/cgi-bin/detail.cgi?id=disease_ontology
- Olivieri, Ricardo (2008), *Implement business logic with the Drools rules engine: Using a declarative programming approach to write your program's business logic*, IBM Research Journal, 18 Mar 2008, Available Online: <http://www.ibm.com/developerworks/java/library/j-drools/>
- Peterson, L. (2009), *K Nearest Neighbor*, Scholarpedia 4(2): 1883
- Popescu, Mihail; Khalilia, Mohammad (2011), *Improving disease prediction using ICD-9 ontological features*, 2011 IEEE International Conference on Fuzzy Systems (FUZZ), pp. 1805 – 1809, Taipei, Taiwan, 27-30 June 2011
- Pauker, S.G.; Szolovits, P. (1978), *Categorical and Probabilistic Reasoning in Medical Diagnosis*, In: Artificial Intelligence Journal, vol. 11, pp. 115-144
- Pazzani, M.J.; Billsus, D. (2007), *Content-based Recommendation Systems*, In LNCS, vol. 4321, pp. 325-341, Brusilovesky, O. (ed.), The adaptive web, Springer-Verlag, May 2007, Available Online: <http://www.fxpal.com/publications/FXPAL-PR-06-383.pdf>
- Qiao Yang Shieh, J.S. (2008), *A multi-agent prototype system for medical diagnosis*, In: 3rd International Conference on Intelligent System and Knowledge Engineering, ISKE 2008, pp. 1265 – 1270, Xiamen, China, 17-19 Nov. 2008
- Quinlan, R. (1999), *Data mining from an AI perspective*, In: PROC INT CONF DATA ENG (ICDE), pp. 186, Sydney, Australia
- Rector, A.L.; Rogers, J.E.; Pole, P. (1996), *The GALEN High Level Ontology*, Proceedings of the Medical Informatics in Europe (MIE), Copenhagen, Denmark

Rector, Alan; Rogers, Jeremy (1999), *Ontological Issues in using a Description Logic to Represent Medical Concepts*, Experience from GALEN, IMIA WORKING GROUP 6 WORKSHOP, Available Online: <http://www.opengalen.org/download/IMIAWG6-1999.pdf>

Reddy, Kiran (2010), *Developing Reliable Clinical Diagnosis Support System*, Available Online: <http://www.kiranreddys.com/articles/clinicaldiagnosissupportsystems.pdf>

Rodriguez, A.; Mencke, M.; Alor-Hernandez, G.; Posada-Gomez, R.; Gomez, J.M.; Aguilar-Lasserre, A.A. (2009), *MEDBOLI: Medical Diagnosis Based on Ontologies and Logical Inference*, International Conference on eHealth, Telemedicine, and Social Medicine, eTELEMED '09, Cancun, Mexico, pp. 233 - 238, 1-7 Feb. 2009

Romero-Tris, Cristina; Riaño, David; Real, Francis (2011), *Ontology-Based Retrospective and Prospective Diagnosis and Medical Knowledge Personalization*, Knowledge Representation for Health-Care, Lecture Notes in Computer Science, vol. 6512, pp. 1-15, Available Online: <http://www.springerlink.com/content/a257522up46ht354/>

Rossi, P. (2003), *Case Management in Health Care*, 2nd Edition, Elsevier, ISBN: 7216-9558-2.

Salem, A.B.M. (2007), *Case Based Reasoning Technology for Medical Diagnosis*, World Academy of Science, Engineering and Technology 31, Available Online: <http://www.waset.org/journals/waset/v31/v31-2.pdf>

Schwimmer, Joshua (2007), *The Value of Mind Mapping*, The Healthline Blog, April 06, 2007, Available Online: http://www.healthline.com/blogs/medical_devices/2007_04_01_medical_devices_archive.html

Scheuermann, R.H.; Werner, C.; Smith, B. (2009), *Toward an Ontological Treatment of Disease and Diagnosis*, Proceedings of the 2009 AMIA Summit on Translational Bioinformatics, pp. 116-120, San Francisco, CA, USA, Available Online: http://ontology.buffalo.edu/medo/Disease_and_Diagnosis.pdf

Schulz, S., *et al.* (2011), *Scalable representations of diseases in biomedical ontologies*, Journal of Biomedical Semantics 2011, vol. 2, issue 2, pp. 6

Shortliffe, E.H. (1987), *Computer programs to support clinical decision making*, In: J. Am. Med. Assoc., vol. 258, pp. 61-66

Shortliffe, E.H.; Buchanan, B.G. (1984), *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, New York, ISBN-10: 0201101726, pp: 769

Shortliffe, E.H.; Teach, R. (1981), *An Analysis of physicians attitudes regarding computer-based consultation systems*, Computer and Biomedical Journal, 14, pp. 542-558, Available Online: http://bmir.stanford.edu/file_asset/index.php/870/BMIR-1981-0052.pdf

Smith, Barry; Brochhausen, Mathias (2011), *Putting biomedical ontologies to work*, Preprint version of paper to appear in Methods of Information in Medicine

Smith, Barry; Werner, Ceusters (2006), *Towards a Reference Terminology for Ontology Research and Development in the Biomedical Domain*, Proceedings of the Second International Workshop on Formal

Biomedical Knowledge Representation: "Biomedical Ontology in Action" (KR-MED 2006), Collocated with the 4th International Conference on Formal Ontology in Information Systems (FOIS-2006), CEUR Workshop Proceedings 222 CEUR-WS.org 2006, Baltimore, Maryland, USA, November 8, 2006, Available Online: http://ontology.buffalo.edu/bfo/Terminology_for_Ontologies.pdf

Soni, Jyoti; Ansari, Ujma; Sharma, Dipesh; Soni, Sunita (2011), *Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction*, International Journal of Computer Applications, vol. 17, issue 8, pp. 43-48, March 2011

Stanford (2011), Available Online: <http://protege.stanford.edu/>

Patel, Chintan; Supekar, Kaustubh; Lee, Yugyung; Park, E.K. (2003), *OntoKhoj: A Semantic Web Portal for Ontology Searching, Ranking and Classification*, In Proc. 5th ACM Int. Workshop on Web Information and Data Management

Polleres, Axel, *et al.* (2009), *XSPARQL Language Specification*, DERI Galway at the National University of Ireland, Galway, Ireland, Available Online: <http://xsparql.deri.org/spec/lang>

Qamar, Rahil (2008), *SEMANTIC MAPPING OF CLINICAL MODEL DATA TO BIOMEDICAL TERMINOLOGIES TO FACILITATE INTEROPERABILITY*, PhD thesis, University of Manchester

Syme, S.L. (2005), *Preventing disease and promoting health: the need for some new thinking*, International Journal of Public Health, Springer-Verlag, vol. 51, issue 5, pp. 247-248

Temple, Larissa; McLeod, R.S.; Gallinger, Steven; Wright, J.G. (2001), *Defining Disease in the Genomics Era*, The Science Journal, vol. 293, issue 5531, pp. 807-808, , 3 August 2001, Available Online: <http://www.sciencemag.org/content/293/5531/807.full>

UK National Innovation Agency (Technology Strategy Board) (2009), *Medicines and Health Care: Executive Summary*, Available Online: <http://www.innovateuk.org/assets/pdf/Corporate-Publications/MedicineHealthcareExecSum.pdf>

University of Maryland (2005), Available Online: http://symptomontologywiki.igs.umaryland.edu/wiki/index.php/Main_Page

Van Niekerk, J.C.; Griffiths, K. (2008), *Advancing Health Care Management with the Semantic Web*, 2008 Third International Conference on Broadband Communications, Information Technology & Biomedical Applications, pp. 373 – 375, Gauteng, South Africa, 23-26 Nov. 2008

Victoria Dept. of Health (2009), *Pathways for Prediabetes, Type 1, Type 2 and Gestational Diabetes*, Department of Health - Loddon Mallee Region, Victoria, Australia

Whetzel, P.L. et al (2010), *Translational Medicine Ontology: A Patient-Centric Ontology for Drug Development and Clinical Practice*, Proceedings of the 2010 AMIA Summit on Translational Bioinformatics, San Francisco, CA, USA, Available Online: http://www.w3.org/wiki/images/4/43/TMO-print_mod.pdf

- Wright, A.; Sittig, D.F. (2008), *A four phase model of the evolution of clinical decision support architectures*, International Journal of Medical Informatics, vol. 77, issue 10, pp. 641-649, 19 March, 2008, Available Online: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2627782/>
- Wiederhold, G.; Shortliffe, E.H.; Fagan, L.; Perreault, L. (2001), *Medical Informatics: Computer Applications in Health Care and Biomedicine*, 2nd Edn., Springer, New York, ISBN-10: 0387984720, pp. 854
- Wiley, Pifer; Williams, *Computer-Aided Diagnosis, Encyclopaedia of Biostatistics* (2005), Published Online, 15 JUL 2005, John Wiley & Sons
- Wilkinson, K.; Sayers, C.; Kuno, H.; Reynolds, D. (2003), *Efficient RDF Storage and Retrieval in Jena 2*, In proceedings of the First International Workshop on Semantic Web and Databases (SWDB), pp. 131-150, Berlin, Germany, Available Online: http://home.zcu.cz/~vcelak/materialy/spravce_sbirek/articles/rdf_storage/efficient_rdf_storage_and_retrieval_in_jena.pdf
- Wim, B.K.; Wiegerinck, W.; Akay, E.; Neijt, J.; Van Beek, A. (2003), *Promedas: A clinical diagnostic decision support system*, Proceedings of the 15th Belgian-Dutch Conference on Artificial Intelligence (BNAIC), Netherlands, Available Online: <http://www.kiranreddys.com/articles/clinicaldiagnosissupportsystems.pdf>
- Witten, Ian H.; Frank, Eibe (2000), *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, San Francisco
- Yang, Q.; Shieh, J.S. (2008), *A multi agent prototype system for medical diagnosis. Intelligent System and Knowledge Engineering*, ISKE 3rd International Conference, vol. 1, pp. 1265-1270, Xiamen, China, Nov. 17-19, 2008
- Yeh, Y.C.; Kuo, Y.H.; Hsu, D.S. (1990), *Building expert systems by embedding analogical reasoning into deductive reasoning mechanism*, In: Ninth Annual International Phoenix Conference on Computers and Communications, pp. 822 – 826, Scottsdale, AZ , USA, 21-23 March 1990
- Ye, Y.; Jiang, Z.; Diao, X.; Yang, D.; Du, G. (2009), *An ontology-based hierarchical semantic modeling approach to clinical pathway workflows*. In: Computers in Biology and Medicine, vol. 39, issue 8, pp. 722-732
- Yuke, Wei; Jiangping, Li (2009), *A Mining Algorithm Study of Obtaining Diagnosis Rule from Clinical Data and Diseases Cases*, 2009 International Conference on Environmental Science and Information Application Technology (ESIAT), vol. 3, pp.411-413, 4-5 July 2009, Wuhan, China
- Zeman1, Martin (2010), *Ontology-Driven Data Preparation for Association Mining*, OntoKDD 2010, Available Online: <http://nb.vse.cz/~svatek/ontoKDD.pdf>
- Zhang, Songmao; Bodenreider, Olivier (2005), *Alignment of multiple ontologies of anatomy: deriving indirect mappings from direct mappings to a reference*, Proceedings for the 2005 AMIA

Appendix A

Evidence-based Clinical Pathways Rules File

```
#created on: 3-Jun-2011
#last modified: 16-Feb-2012
package medRules.test

#list any import classes here.

import medRules.test.Disease;
import medRules.test.Test;
import medRules.test.TestResult;
import java.util.LinkedList;
import java.util.List;

#declare any global variables here

rule "1 - If a fasting plasma glucose test is performed to diagnose Diabetes,
then a result 109.8 mg/dL or less indicates negative diagnosis or normal
glucose levels"

when

    t:Test(testName == "Fasting plasma glucose")
    d:Disease(diseaseName == "diabetes mellitus" || diseaseName ==
"diabetes mellitus type 2"
        || diseaseName == "diabetes mellitus type 1" || diseaseName ==
"diabetes" || diseaseName == "Diabetes")

    tr:TestResult(id == 1 && unit == "mg/dL" && description == "glucose -
FPG" && amount <= 109.8 && amount > 0)

then
    tr.setNormal(true);
    tr.setDiagnosis(false);
    tr.setResult("Normal healthy glucose levels, no evidence of diabetes");
end

rule "2 - If a fasting plasma glucose test is performed to diagnose Diabetes,
then a result 109.8-125 mg/dL indicates positive pre-diabetes"

when

    t:Test(testName == "Fasting plasma glucose")
    d:Disease(diseaseName == "diabetes mellitus" || diseaseName ==
"diabetes mellitus type 2"
        || diseaseName == "diabetes mellitus type 1" || diseaseName ==
"diabetes" || diseaseName == "Diabetes")

    tr:TestResult(id == 1 && unit == "mg/dL" && description == "glucose -
FPG" && amount > 109.8 && amount <= 125)
```

then

```
tr.setNormal(false);
tr.setDiagnosis(false);
tr.setResult("Abnormal glucose levels indicate a state of pre-diabetes
and increased risk of developing type 2 diabetes");
end
```

rule "3 - When a fasting plasma glucose test is performed to diagnose Diabetes, then a result higher than 125 mg/dL or higher indicates positive diabetes"

when

```
t:Test(testName == "Fasting plasma glucose")
d:Disease(diseaseName == "diabetes mellitus" || diseaseName ==
"diabetes mellitus type 2"
|| diseaseName == "diabetes mellitus type 1" || diseaseName ==
"diabetes" || diseaseName == "Diabetes")
```

```
tr:TestResult(id == 1 && unit == "mg/dL" && description == "glucose -
FPG" && amount > 125)
```

then

```
tr.setNormal(false);
tr.setDiagnosis(true);
tr.setResult("positive test for diabetes");
```

end

rule "4 - If an oral glucose tolerance test is performed to diagnose Diabetes, then a result of less than 140 mg/dL indicates negative pre-diabetes"

when

```
t:Test(testName == "Oral glucose tolerance")
d:Disease(diseaseName == "diabetes mellitus" || diseaseName ==
"diabetes mellitus type 2"
|| diseaseName == "diabetes mellitus type 1" || diseaseName ==
"diabetes" || diseaseName == "Diabetes")
```

```
tr:TestResult(id == 2 && unit == "mg/dL" && description == "glucose -
OGGT" && amount <= 140 && amount > 0)
```

then

```
tr.setNormal(true);
tr.setDiagnosis(false);
tr.setResult("Normal healthy glucose levels, no evidence of diabetes");
```

end

rule "5 - If an oral glucose tolerance test is performed to diagnose Diabetes, then a result of 140-200 mg/dL indicates positive pre-diabetes"

when

```
t:Test(testName == "Oral glucose tolerance")
```

```

    d:Disease(diseaseName == "diabetes mellitus" || diseaseName ==
"diabetes mellitus type 2"
        || diseaseName == "diabetes mellitus type 1" || diseaseName ==
"diabetes" || diseaseName == "Diabetes")
    tr:TestResult(id == 2 && unit == "mg/dL" && description == "glucose -
OGGT" && amount > 140 && amount <= 200)

then
    tr.setNormal(false);
    tr.setDiagnosis(false);
    tr.setResult("Abnormal glucose levels indicate a state of pre-diabetes
and increased risk of developing type 2 diabetes");
end

rule "6 - If an oral glucose tolerance test is performed to diagnose Diabetes,
then a result of of 200 mg/dL or higher indicates a positive diagnosis of
diabetes"

when

    t:Test(testName == "Oral glucose tolerance")
    d:Disease(diseaseName == "diabetes mellitus" || diseaseName ==
"diabetes mellitus type 2"
        || diseaseName == "diabetes mellitus type 1" || diseaseName ==
"diabetes" || diseaseName == "Diabetes")
    tr:TestResult(id == 2 && unit == "mg/dL" && description == "glucose -
OGGT" && amount > 200)

then
    tr.setNormal(false);
    tr.setDiagnosis(true);
    tr.setResult("High glucose levels indicate diabetes");
end

rule "7 - If a blood pressure test is performed to diagnose hypertension,
then a result of 60-80 mmHg diastolic blood pressure would signal normal bp"

when

    d:Disease(diseaseName == "Hypertension" || diseaseName ==
"hypertension")
    t:Test(testName == "Blood pressure" || testName == "blood pressure")
    tr:TestResult(id == 5 && unit == "mmHg" && amount <= 80 && (description
== "Diastolic" || description == "diastolic") && amount >= 60)

then

    tr.setNormal(true);
    tr.setDiagnosis(false);
    tr.setResult("Normal healthy diastolic bp, no indication of
hypertension");
end

rule "8 - If a blood pressure test is performed to diagnose hypertension,
then a result of more than 90 mmHg diastolic blood pressure would signal
hypertension"

```


when

```
d:Disease(diseaseName == "Hypertension" || diseaseName ==
"hypertension")
  t:Test(testName == "Blood pressure" || testName == "blood pressure")
  tr:TestResult(id == 5 && unit == "mmHg" && amount >= 90 &&
(description == "Diastolic" || description == "diastolic"))
```

then

```
tr.setNormal(false);
tr.setDiagnosis(true);
tr.setResult("hypertension - high diastolic blood pressure");
```

end

rule "9 - If a blood pressure test is performed to diagnose hypertension, then a result of 90-120 mmHg systolic blood pressure would signal normal bp"

when

```
d:Disease(diseaseName == "Hypertension" || diseaseName ==
"hypertension")
  t:Test(testName == "Blood pressure" || testName == "blood pressure")
  tr:TestResult(id == 4 && unit == "mmHg" && amount <= 120 &&
(description == "systolic" || description == "Systolic") && amount >= 90)
```

then

```
tr.setNormal(true);
tr.setDiagnosis(false);
tr.setResult("Normal healthy systolic bp, no indication of
hypertension");
```

end

rule "10 - If a blood pressure test is performed to diagnose hypertension, then a result of more than 140 mmHg systolic blood pressure would signal hypertension"

when

```
d:Disease(diseaseName == "Hypertension" || diseaseName ==
"hypertension")
  t:Test(testName == "Blood pressure" || testName == "blood pressure")
  tr:TestResult(id == 4 && unit == "mmHg" && amount >= 140 &&
(description == "systolic" || description == "Systolic"))
```

then

```
tr.setNormal(false);
tr.setDiagnosis(true);
tr.setResult("hypertension - high systolic blood pressure");
```

end

rule "11 - If a blood pressure test is performed to diagnose hypertension, then a result between 80-89 mmHg diastolic blood pressure would indicate pre-hypertension"

when

```
d:Disease(diseaseName == "Hypertension" || diseaseName ==
"hypertension")
  t:Test(testName == "Blood pressure" || testName == "blood pressure")
  tr:TestResult(id == 5 && unit == "mmHg" && amount > 80 && amount <= 89
&& (description == "Diastolic" || description == "diastolic") && amount > 0)
```

then

```
tr.setNormal(false);
tr.setDiagnosis(false);
tr.setResult("pre-hypertension, diastolic bp not normal");
```

end

rule "12 - If a blood pressure test is performed to diagnose hypertension, then a result between 120-139 mmHg systolic blood pressure would indicate pre-hypertension"

when

```
d:Disease(diseaseName == "Hypertension" || diseaseName ==
"hypertension")
  t:Test(testName == "Blood pressure" || testName == "blood pressure")
  tr:TestResult(id == 4 && unit == "mmHg" && amount >= 120 && amount <=
139 && (description == "systolic" || description == "Systolic"))
```

then

```
tr.setNormal(false);
tr.setDiagnosis(false);
tr.setResult("pre-hypertension, systolic bp not normal");
```

end

rule "13 - If an arterial blood gas test is performed to diagnose adult respiratory distress syndrome, then a result between 75-100 PaO2 is normal"

when

```
d:Disease(diseaseName == "adult respiratory distress syndrome" ||
diseaseName == "Adult respiratory distress syndrome")
  t:Test(testName == "Arterial blood gas" || testName == "arterial blood
gas")
  tr:TestResult(id == 6 && unit == "PaO2" && amount >= 75 && amount <=
100)
```

then

```
tr.setNormal(true);
tr.setDiagnosis(false);
```

```
tr.setResult("normal PaO2, no indication of adult respiratory distress syndrome");
```

end

rule "14 - If an arterial blood gas test is performed to diagnose adult respiratory distress syndrome, then a result below 75 PaO2 indicates diagnosis of Adult respiratory distress syndrome"

when

```
d:Disease(diseaseName == "adult respiratory distress syndrome" ||
diseaseName == "Adult respiratory distress syndrome")
t:Test(testName == "Arterial blood gas" || testName == "arterial blood
gas")
tr:TestResult(id == 6 && unit == "PaO2" && amount < 75 && amount > 0)
```

then

```
tr.setNormal(false);
tr.setDiagnosis(true);
tr.setResult("Adult respiratory distress syndrome");
```

end

rule "15 - If a CBC (Complete Blood Count) test is performed to diagnose Anemia, then a count of 38.8-50 % of red blood cells in the blood for males indicates normal rbc levels"

when

```
d:Disease(diseaseName == "Anemia" || diseaseName == "anemia")
t:Test(testName == "Complete blood count" || testName == "CBC")
tr:TestResult(id == 9 && unit == "%" && description == "% RBC, male" &&
amount >= 38.8 && amount <= 50)
```

then

```
tr.setNormal(true);
tr.setDiagnosis(false);
tr.setResult("normal percentage of red blood cells in blood, negative
test for anemia");
```

end

rule "16 - If a CBC (Complete Blood Count) test is performed to diagnose Anemia, then a count of 34.9-44.5 % of red blood cells in the blood for females indicates normal rbc levels"

when

```
d:Disease(diseaseName == "Anemia" || diseaseName == "anemia")
t:Test(testName == "Complete blood count" || testName == "CBC")
tr:TestResult(id == 10 && unit == "%" && description == "% RBC, female"
&& amount >= 34.9 && amount <= 44.5)
```

then

```
tr.setNormal(true);  
tr.setDiagnosis(false);  
tr.setResult("normal percentage of red blood cells in blood, negative  
test for anemia");
```

end

rule "17 - If a CBC (Complete Blood Count) test is performed to diagnose Anemia, then 13.5-17.5 mg/dL of hemoglobin the blood for males indicates normal rbc levels"

when

```
d:Disease(diseaseName == "Anemia" || diseaseName == "anemia")  
t:Test(testName == "Complete blood count" || testName == "CBC")  
tr:TestResult(id == 8 && unit == "mg/dL" && description == "hemoglobin,  
male" && amount >= 13.5 && amount <= 17.5)
```

then

```
tr.setNormal(true);  
tr.setDiagnosis(false);  
tr.setResult("normal hemoglobin levels in blood, negative test for  
anemia");
```

end

rule "18 - If a CBC (Complete Blood Count) test is performed to diagnose Anemia, then 12-15.5 mg/dL of hemoglobin the blood for females indicates normal rbc levels"

when

```
d:Disease(diseaseName == "Anemia" || diseaseName == "anemia")  
t:Test(testName == "Complete blood count" || testName == "CBC")  
tr:TestResult(id == 7 && unit == "mg/dL" && description == "hemoglobin,  
female" && amount >= 12 && amount <= 15.5)
```

then

```
tr.setNormal(true);  
tr.setDiagnosis(false);  
tr.setResult("normal hemoglobin levels in blood, negative test for  
anemia");
```

end

rule "19 - If a CBC (Complete Blood Count) test is performed to diagnose Anemia, then a count of less than 38.8 % of red blood cells in the blood for males indicates anemia"

when

```
d:Disease(diseaseName == "Anemia" || diseaseName == "anemia")  
t:Test(testName == "Complete blood count" || testName == "CBC")
```

```
tr:TestResult(id == 9 && unit == "%" && description == "% RBC, male" &&
amount < 38.8 && amount > 0)
```

then

```
tr.setNormal(false);
tr.setDiagnosis(true);
tr.setResult("percentage of red blood cells in blood indicates
anemia");
```

end

rule "20 - If a CBC (Complete Blood Count) test is performed to diagnose Anemia, then a count of less than 34.9 % of red blood cells in the blood for females indicates anemia"

when

```
d:Disease(diseaseName == "Anemia" || diseaseName == "anemia")
t:Test(testName == "Complete blood count" || testName == "CBC")
tr:TestResult(id == 10 && unit == "%" && description == "% RBC, female"
&& amount < 34.9 && amount > 0)
```

then

```
tr.setNormal(false);
tr.setDiagnosis(true);
tr.setResult("low percentage of red blood cells in blood indicates
anemia");
```

end

rule "21 - If a CBC (Complete Blood Count) test is performed to diagnose Anemia, then less than 13.5 mg/dL of hemoglobin the blood for males indicates anemia"

when

```
d:Disease(diseaseName == "Anemia" || diseaseName == "anemia")
t:Test(testName == "Complete blood count" || testName == "CBC")
tr:TestResult(id == 8 && unit == "mg/dL" && description == "hemoglobin,
male" && amount < 13.5 && amount > 0)
```

then

```
tr.setNormal(false);
tr.setDiagnosis(true);
tr.setResult("low hemoglobin levels in blood indicate anemia");
```

end

rule "22 - If a CBC (Complete Blood Count) test is performed to diagnose Anemia, then less than 12 mg/dL of hemoglobin the blood for females indicates anemia"

when

```
d:Disease(diseaseName == "Anemia" || diseaseName == "anemia")
t:Test(testName == "Complete blood count" || testName == "CBC")
tr:TestResult(id == 7 && unit == "mg/dL" && description == "hemoglobin,
female" && amount < 12 && amount > 0)
```

then

```
tr.setNormal(false);
tr.setDiagnosis(true);
tr.setResult("low hemoglobin levels in blood indicate anemia");
```

end

rule "23 - If a blood test is performed to diagnose calcemia, then 9-10.5 mg/dL of calcium in the blood indicates normal calcium levels"

when

```
d:Disease(diseaseName == "Calcemia" || diseaseName == "calcemia" ||
diseaseName == "hypercalcemia" || diseaseName == "Hypercalcemia")
t:Test(testName == "blood test" || testName == "Blood test")
tr:TestResult(id == 11 && unit == "mg/dL" && description == "calcium
levels" && amount >= 9 && amount <= 10.5)
```

then

```
tr.setNormal(true);
tr.setDiagnosis(false);
tr.setResult("normal calcium levels in blood, negative test for
hypercalcemia");
```

end

rule "24 - If a blood test is performed to diagnose calcemia, then more than 10.5 mg/dL of calcium in the blood indicates calcemia"

when

```
d:Disease(diseaseName == "Calcemia" || diseaseName == "calcemia" ||
diseaseName == "hypercalcemia" || diseaseName == "Hypercalcemia")
t:Test(testName == "blood test" || testName == "Blood test")
tr:TestResult(id == 11 && unit == "mg/dL" && description == "calcium
levels" && amount > 10.5)
```

then

```
tr.setNormal(false);
tr.setDiagnosis(true);
tr.setResult("calcemia - elevated/high calcium levels in the blood");
```

end

rule "25 - If a blood pressure test is performed to diagnose hypertension, then a result of 60-90 mmHg systolic blood pressure would signal hypotension"

when

```
    d:Disease(diseaseName == "Hypertension" || diseaseName ==
"hypertension")
    t:Test(testName == "Blood pressure" || testName == "blood pressure")
    tr:TestResult(id == 4 && unit == "mmHg" && amount <= 90 && (description
== "systolic" || description == "Systolic") && amount >= 60)
```

then

```
    tr.setNormal(false);
    tr.setDiagnosis(true);
    tr.setResult("Low systolic bp indicates hypotension");
```

end

rule "26 - If a blood pressure test is performed to diagnose hypertension, then a result of 40-60 mmHg diastolic blood pressure would signal hypotension"

when

```
    d:Disease(diseaseName == "Hypertension" || diseaseName ==
"hypertension")
    t:Test(testName == "Blood pressure" || testName == "blood pressure")
    tr:TestResult(id == 5 && unit == "mmHg" && amount >= 40 && (description
== "Diastolic" || description == "diastolic") && amount <= 60)
```

then

```
    tr.setNormal(true);
    tr.setDiagnosis(false);
    tr.setResult("Low diastolic bp indicates hypotension");
```

end

rule "27 - If a blood test is performed to diagnose calcemia, then 7-9 mg/dL of calcium in the blood indicates low calcium levels"

when

```
    d:Disease(diseaseName == "Calcemia" || diseaseName == "calcemia" ||
diseaseName == "hypercalcemia" || diseaseName == "Hypercalcemia")
    t:Test(testName == "blood test" || testName == "Blood test")
    tr:TestResult(id == 11 && unit == "mg/dL" && description == "calcium
levels" && amount >= 7 && amount <= 9)
```

then

```
    tr.setNormal(false);
    tr.setDiagnosis(true);
    tr.setResult("low calcium levels in blood, positive test for
hypocalcemia");
```

end

Using Weka GUI toolkit for Classification and Association

Using Weka, one can perform some essential pre-processing steps on the data using the pre-processing tab (See Figure A1).

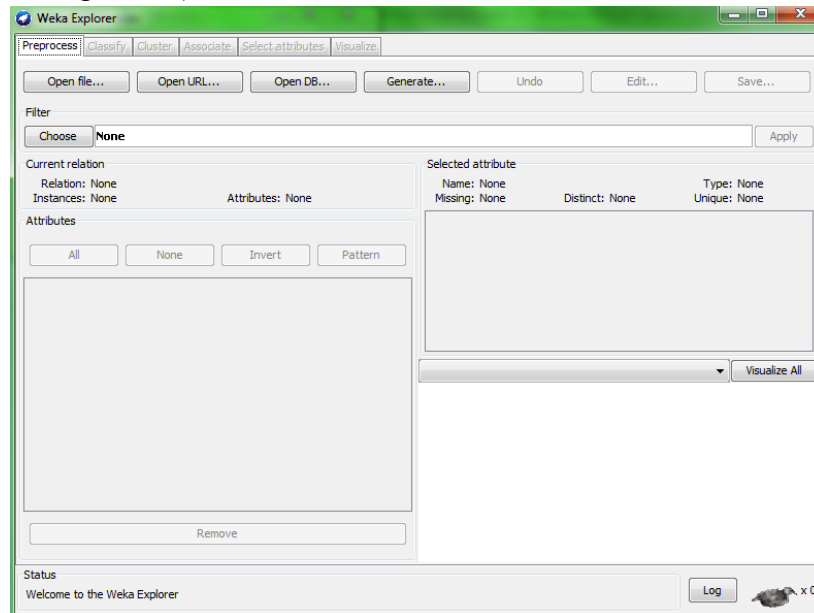


Fig. A1: Weka GUI Pre-processing Tab

The **Preprocess tab** allows the user to enter data into the tool in several formats, apply various filters to the data, and visualize the data. One filter we will be using is the **Discretize** filter. To select the **Discretize** filter, click the **Choose** button (see figures A2 & A3 below), select **filters** → **unsupervised** → **attribute** → **Discretize**

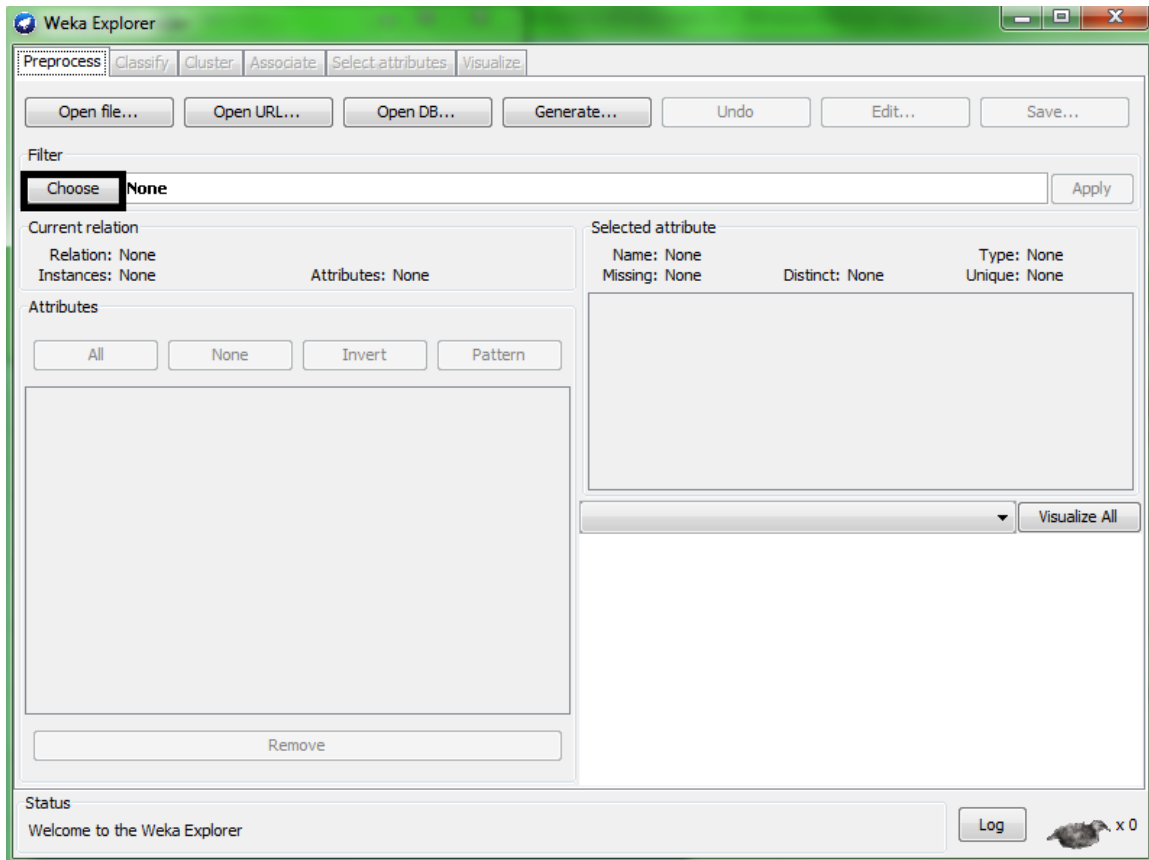


Fig. A2: Choose Filter Button in the Weka GUI Pre-processing Tab

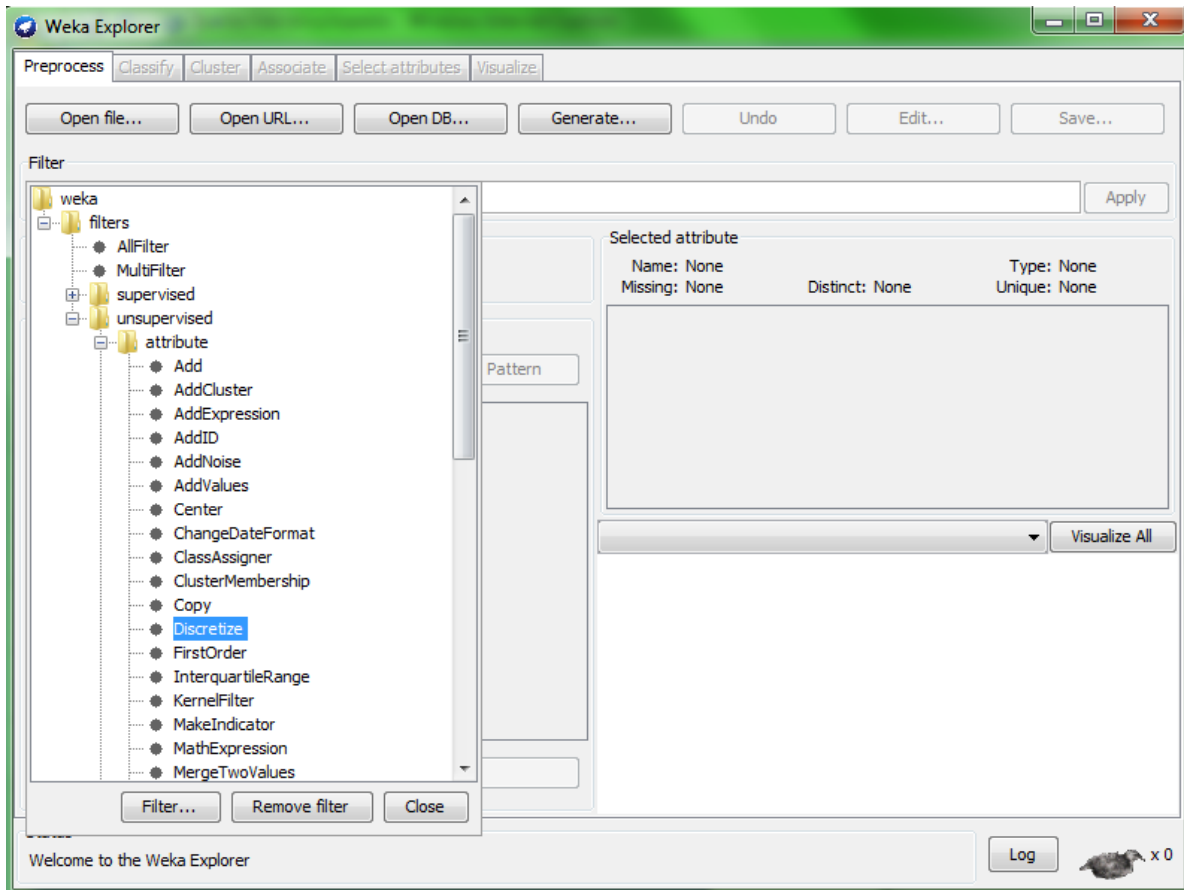


Fig. A3: Discretize Filter Button in Weka

Next, we open the NIS diabetes data file (Table 4.8) to data mining operations. Once the data is imported into the tool, the **Preprocess** tab shows several things about the data. In the figure below (figure A4), the display inside the black box shows the attributes and class variable of the diabetes data. The blue box displays statistics about the values of the selected attribute in the black box. The red is a bar graph of the values of the selected variable. In this case, the selected variable is called *fasting_plasma_glucose*. It is a test used to determine glucose levels in the blood. The bars are colour coded to show the correlation/relationship between the values of the attribute and the value of the class variable or classification. In this case, the class variable is *prim_diagnosis* short for primary diagnosis. It has two values in this data either positive diagnosis of type 2 (code blue) diabetes or no indication of type 2 diabetes (code red). Here, all the data records with a *fasting_plasma_glucose* level in the range 55-91 mg/dL are coded red or no indication of type 2 diabetes. Also, all the patient records in the range of 125-161 mg/dL meaning all these patients have type 2. This means is the *fasting_plasma_glucose* is very indicative of whether a patient has type 2 diabetes or not.

diabetes.

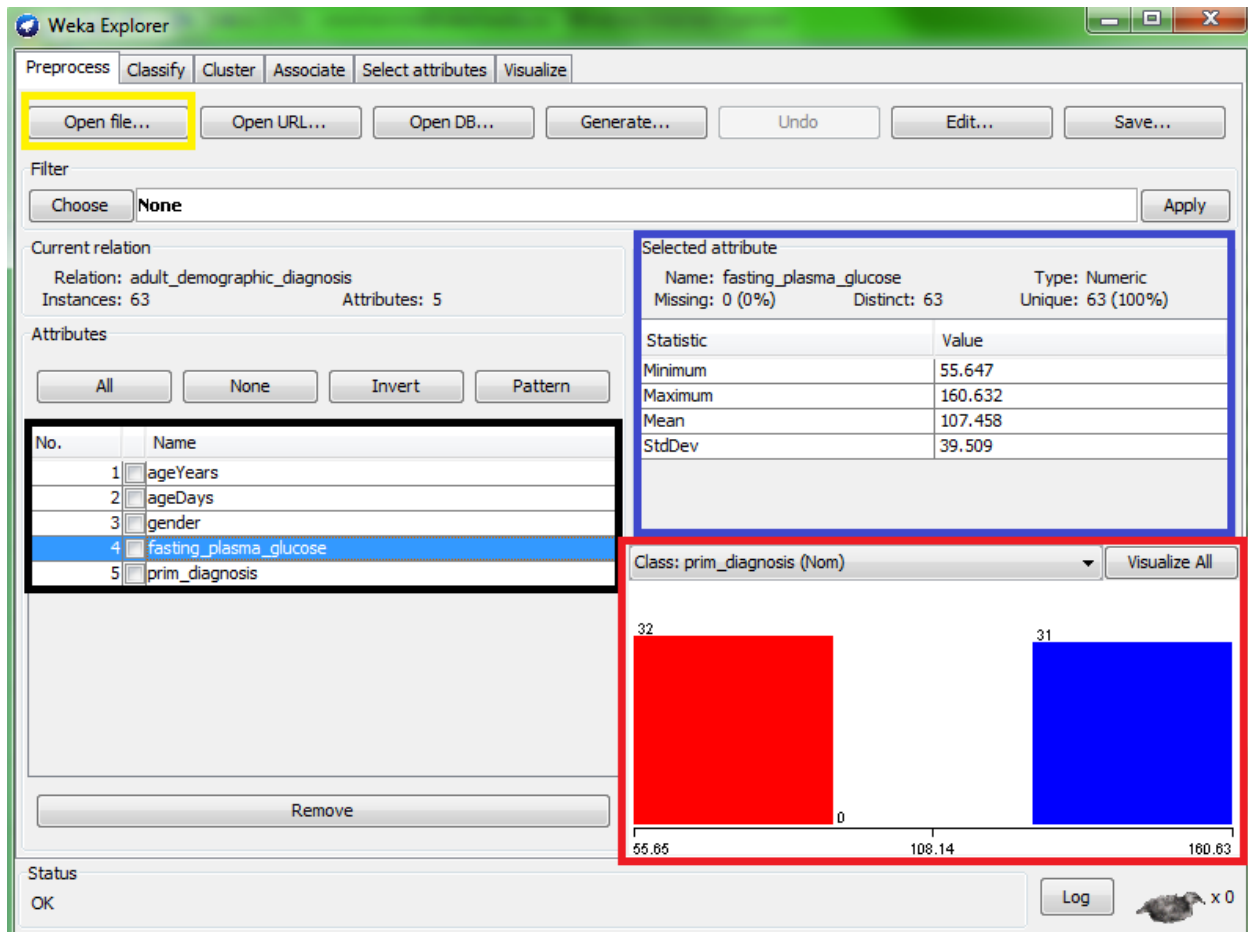


Fig. A4: Diabetes Data 1 Statistics in the Weka Pre-processing Tab

Using Weka for Classification

In this section we are demonstrating how one can use Weka for classification to predict the value of the class attribute based on the values of the other given training attributes. For this purpose, we need first to use one of the classification algorithms provided by Weka. We are going to use the data provided in table 4.8. In this classification example, we aim to determine whether patients have type 2 diabetes (value of class attribute) or not based on the values of attributes *age*, *gender*, and *fasting_plasma_glucose*. Using Weka, the **Classify** tab provides access to a variety of classification algorithms ranging from bayes algorithms to rule-based algorithms to lazy algorithms to tree-based algorithms. In this example, we will go through the process of classification using the J48 Java implementation of the C4.5 decision tree classification algorithm. Then we will tabulate results for other classification algorithms. So let us start by using the **Classify** tab and then selecting select the **Choose** button (see figure A5).

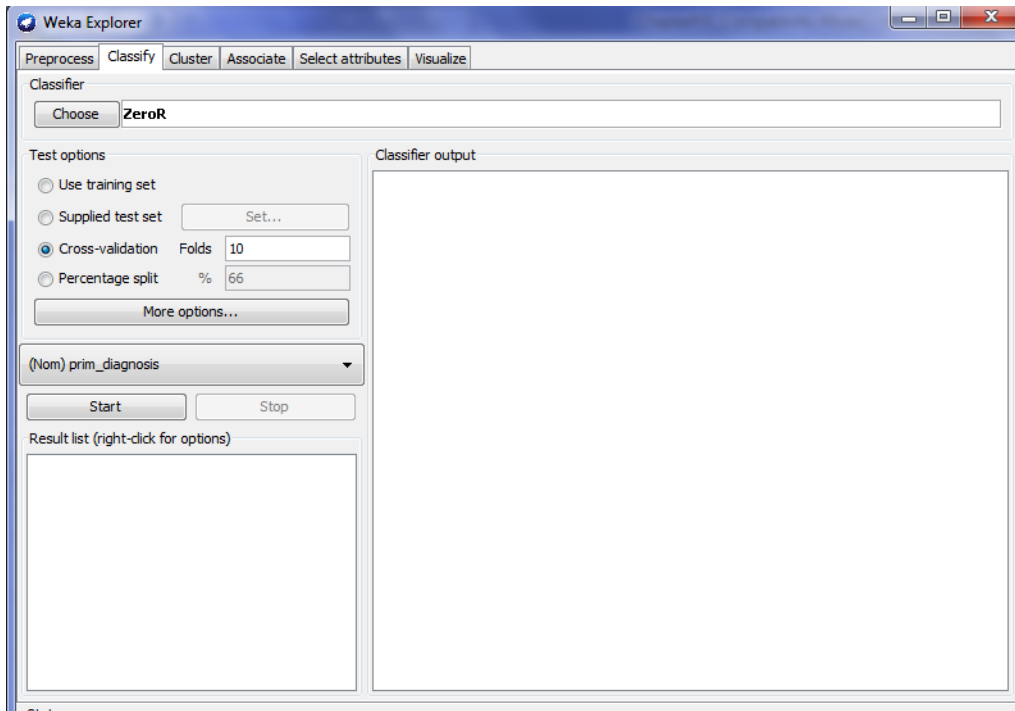


Fig. A5: Weka GUI Classify Tab

Then, we select the J48 tree based algorithm by selecting **trees**→**J48** (figure A6)

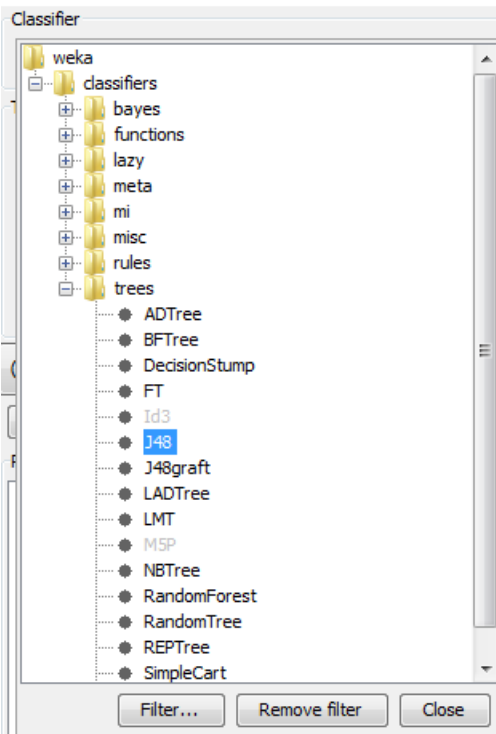


Fig. A6: Weka J48 Classification Algorithm

Next, under **Test options**, we select **Use training set**, and then click **Start** (figure A7)

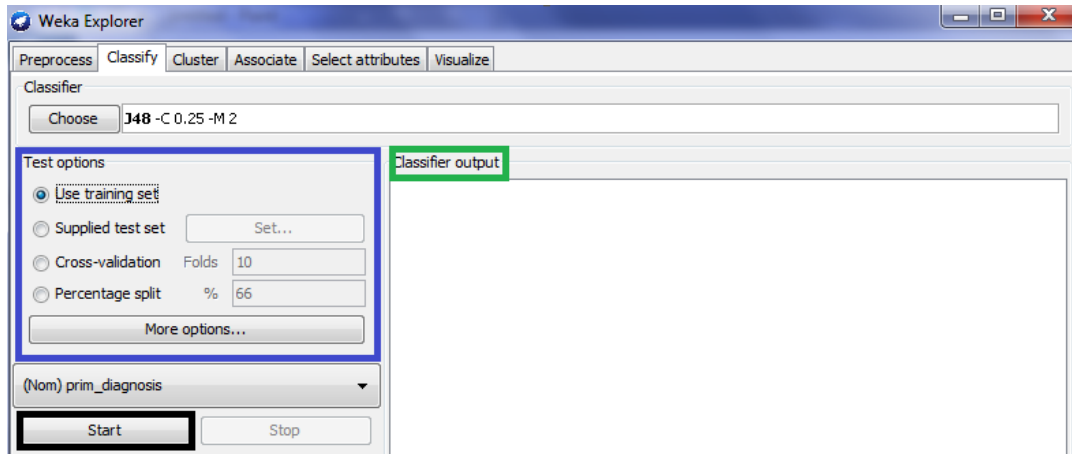


Fig. A7: Weka GUI Training Set Option in the Classify Tab

The **Classifier output** window will report the following results (figure A8):

```

J48 pruned tree
-----
fasting_plasma_glucose <= 89.552319: Normal (32.0)
fasting_plasma_glucose > 89.552319: DIABETES UNCOMPL TYPE II (31.0)

Number of Leaves :    2
Size of the tree :    3

Time taken to build model: 0seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances      63      100    %
Incorrectly Classified Instances    0        0    %
Kappa statistic                     1
Mean absolute error                  0
Root mean squared error              0
Relative absolute error              0    %
Root relative squared error          0    %
Total Number of Instances          63

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      1      0      1      1      1      1      DIABETES UNCOMPL TYPE II
      1      0      1      1      1      1      Normal
Weighted Avg.  1      0      1      1      1      1

=== Confusion Matrix ===
 a  b  <-- classified as
31  0  |  a = DIABETES UNCOMPL TYPE II
 0 32  |  b = Normal

```

Fig. A8: Classifier Output Window in the Weka GUI Classify Tab showing results of running the J48 Classification Algorithm on Table 4.8

In the green box, the pruned J48 decision tree is shown. A graph of this tree is below in figure A9.

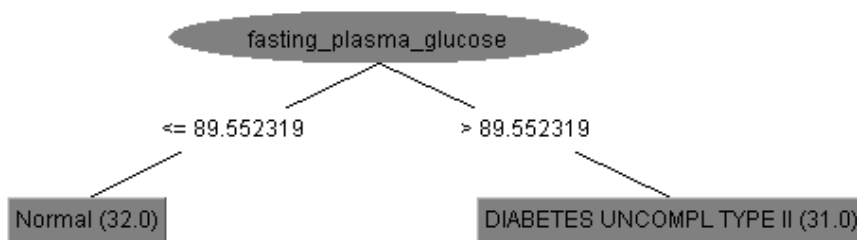


Fig. A9: J48 Decision Tree for Diabetes Data 1 Classification of the value of the Primary Diagnosis Class Variable

The blue box shows the accuracy of the classification. In this case, all (100%) 63 instances/records were correctly classified as uncomplicated type 2 diabetes or normal. The green shows the confusion matrix of the classification. The matrix has four entries as follows:

[Diabetes CA Diabetes, Diabetes CA Normal; Normal CA Diabetes, Normal CA Normal]
where CA stands for classified as.

To verify the accuracy of the classification, it is possible to use a number of test patient records where the value of the class variable is unknown to the classification algorithm.

In our example, a typical test record looks like this:

Age, Gender, FPG, Primary_Diagnosis
32, 1.0, 87, ?

The value of the class attribute is unknown and is predicted by the classification algorithm based on the value of the other attributes. The classification algorithm uses the knowledge it gained from the data where the value of the class variable is known (called training data) to predicted the value of the class variable for the test data (and assign the value instead of the question mark). Usually, the value of the class attribute is known but not given to the classification algorithm so the classification algorithm can predict the value and these predictions can be compared to the actual values. Prediction accuracy can then be recorded and accuracy of prediction of several algorithms on a data set can be compared.

Under **Test options**, we select **Supplied test set** and click **Set...**, then select the file with the test records, finally click **Start** (see figure A10 below)

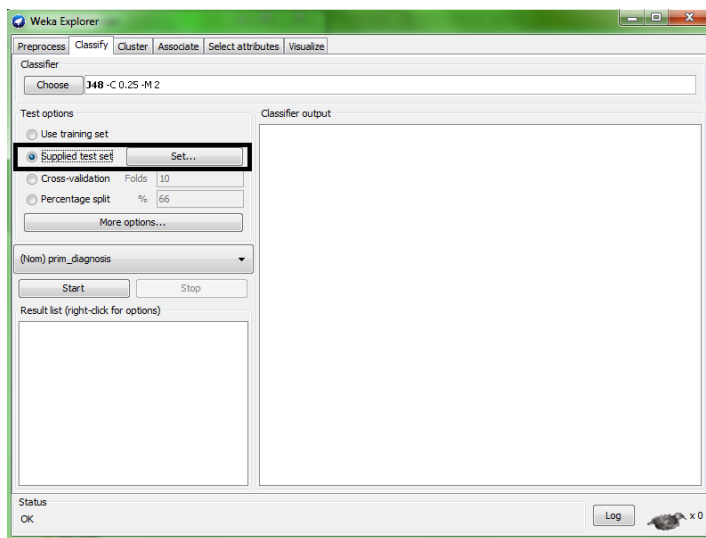


Fig. A10: Classifying Test Records using the Weka Classify Tab

Next the results of classification can be saved to a file. The saved results can then be compared to the actual values.

In our example, the following represents the test data (figure A11):

```
% 1. Age in Years
% 2. Age in Days (if age in years = 0)
% 3. Patient's Gender (0 Male, 1 Female, NaN Unknown)
% 4. Fasting plasma glucose level in the blood
% 5. Primary Diagnosis

@relation adult_demographic_diagnosis
@attribute 'ageYears' real
@attribute 'ageDays' real
@attribute 'gender' {0.0, 1.0, NaN}
@attribute 'fasting_plasma_glucose' real
@attribute 'prim_diagnosis' {"DIABETES UNCOMPL TYPE II", "Normal"}
@data
36.0, 0.0, 0.0, 69.7002091002019, ?
80.0, 0.0, 1.0, 136.2817563296179, ?
95.0, 0.0, 1.0, 64.41565152498089, ?
54.0, 0.0, 0.0, 160.39067235821383, ?
51.0, 0.0, 0.0, 71.20177736456496, ?
37.0, 0.0, 0.0, 142.67747545000253, ?
33.0, 0.0, 1.0, 64.79464064303514, ?
32.0, 0.0, 0.0, 157.5341087879713, ?
40.0, 0.0, 0.0, 55.646933848535824, ?
51.0, 0.0, 1.0, 156.18428598232236, ?
23.0, 0.0, 1.0, 68.28715336384086, ?
```

Fig. A11: Diabetes Test Records in ARFF Format

Note that the primary diagnosis is unknown here. These are the test records where the algorithm needs to predict primary diagnosis. After the J48 classification algorithm runs, the predicted values for primary diagnosis are saved to the file below (figure A12):

```
@relation adult_demographic_diagnosis_predicted

@attribute ageYears numeric
@attribute ageDays numeric
@attribute gender {0.0,1.0,NaN}
@attribute fasting_plasma_glucose numeric
@attribute predictedprim_diagnosis {'DIABETES UNCOMPL TYPE II','Normal'}

@data
36,0,0.0,69.700209,Normal
80,0,1.0,136.281756,'DIABETES UNCOMPL TYPE II'
95,0,1.0,64.415652,Normal
54,0,0.0,160.390672,'DIABETES UNCOMPL TYPE II'
51,0,0.0,71.201777,Normal
37,0,0.0,142.677475,'DIABETES UNCOMPL TYPE II'
33,0,1.0,64.794641,Normal
32,0,0.0,157.534109,'DIABETES UNCOMPL TYPE II'
40,0,0.0,55.646934,Normal
51,0,1.0,156.184286,'DIABETES UNCOMPL TYPE II'
23,0,1.0,68.287153,Normal
```

Fig. A12: Predicted Classification of Diabetes Test Records in ARFF Format

The actual results for primary diagnosis for the test results were previously saved into a third file (figure A13):

```

% 1. Age in Years
% 2. Age in Days (if age in years = 0)
% 3. Patient's Gender (0 Male, 1 Female, NaN Unknown)
% 4. Fasting plasma glucose level in the blood
% 5. Primary Diagnosis

@relation adult_demographic_diagnosis
@attribute 'ageYears' real
@attribute 'ageDays' real
@attribute 'gender' {0.0, 1.0, NaN}
@attribute 'fasting_plasma_glucose' real
@attribute 'prim_diagnosis' {"DIABETES UNCOMPL TYPE II", "Normal"}
@data
36.0, 0.0, 0.0, 69.7002091002019, "Normal"
80.0, 0.0, 1.0, 136.2817563296179, "DIABETES UNCOMPL TYPE II"
95.0, 0.0, 1.0, 64.41565152498089, "Normal"
54.0, 0.0, 0.0, 160.39067235821383, "DIABETES UNCOMPL TYPE II"
51.0, 0.0, 0.0, 71.20177736456496, "Normal"
37.0, 0.0, 0.0, 142.67747545000253, "DIABETES UNCOMPL TYPE II"
33.0, 0.0, 1.0, 64.79464064303514, "Normal"
32.0, 0.0, 0.0, 157.5341087879713, "DIABETES UNCOMPL TYPE II"
40.0, 0.0, 0.0, 55.646933848535824, "Normal"
51.0, 0.0, 1.0, 156.18428598232236, "DIABETES UNCOMPL TYPE II"
23.0, 0.0, 1.0, 68.28715336384086, "Normal"

```

Fig. A13: Actual Classification of Diabetes Test Records in ARFF Format

Using a program, the actual results are then compared with the predicted results and the accuracy of prediction is calculated. In our example, using the J48 algorithm the accuracy is 100%. Using Weka we can use other classification algorithms that may provide different accuracies and results. The JRIP rule-based classification algorithm would also give 100% accuracy in this example. JRIP algorithm produces rules that fit the data which may be of interest to the clinician, and assigns confidence odds/probabilities for these rules. The confidence factor indicates JRIP's confidence in the rule. The rule found by JRIP found to apply to this data is listed below. It states that if *fasting_plasma_glucose* is ≥ 127.914274 then primary diagnosis is Type 2 Diabetes with 100% confidence (odds indicated in brackets-see rule below); otherwise the diagnosis is normal or no indication of Type 2 Diabetes with 100% confidence.

JRIP

JRIP rules:

=====

```

(fasting_plasma_glucose >= 127.914274) => prim_diagnosis=DIABETES UNCOMPL TYPE II (31.0/0.0)
=> prim_diagnosis=Normal (32.0/0.0)

```

Another algorithm that indicates 100% accuracy with this data is NaiveBayes. NaiveBayes outputs classification statistics for the data. See figure A14 below. Notice under *gender*, 17 males were diagnosed with diabetes and 11 were normal. In comparison, 16 females were diagnosed with diabetes and 23 were normal. This suggests Type 2 diabetes is more common among males. Also note under *fasting_plasma_glucose*, the glucose levels for those with Type 2 diabetes have a mean of 146 mg/dL (well into the diabetes region), while the glucose levels for the normal cases have a mean of 70 mg/dL which falls into the normal range for an FPG test.

Naive Bayes

Naive Bayes Classifier

Attribute	Class	
	DIABETES UNCOMPL TYPE II (0.49)	Normal (0.51)
=====		
<i>ageYears</i>		
mean	30.0951	32.1234
std. dev.	20.7298	29.4547
weight sum	31	32
precision	2.4359	2.4359
<i>ageDays</i>		
mean	0	2.3125
std. dev.	3.0833	8.9563
weight sum	31	32
precision	18.5	18.5
<i>gender</i>		
0.0	17.0	11.0
1.0	16.0	23.0
NaN	1.0	1.0
[total]	34.0	35.0
<i>fasting_plasma_glucose</i>		
mean	146.007	70.0078
std. dev.	10.2929	9.1149
weight sum	31	32
precision	1.6933	1.6933

Fig. A14: Results of the Naive Bayes Classification Algorithm when Applied to Diabetes Data 1 Data Table

The NBTree is another algorithm which indicates 100% accuracy for the diabetes data. This algorithm is a decision tree version of the NaiveBayes algorithm. It produces similar output. See figure C15 below.

NBTree

```

NBTree
-----
: NBO

Leaf number: 0 Naive Bayes Classifier

Attribute                Class
                        DIABETES UNCOMPL TYPE II
                        (0.49)
                        Normal
                        (0.51)
=====
ageYears
  'All'                   32.0                   33.0
  [total]                 32.0                   33.0

ageDays
  'All'                   32.0                   33.0
  [total]                 32.0                   33.0

gender
  0.0                     17.0                   11.0
  1.0                     16.0                   23.0
  NaN                      1.0                    1.0
  [total]                 34.0                   35.0

fasting_plasma_glucose
  '(-inf-108.733297]'    1.0                    33.0
  '(108.733297-inf)'    32.0                    1.0
  [total]                 33.0                   34.0

```

Fig. C15: Results of the NBTree Classification Algorithm when Applied to Diabetes Data 1 Data Table

Now we will present, below in table C1, the accuracy of prediction figures for different classification algorithms for the above NIS diabetes data, and for the other NIS data tables (Tables B1, B2, and B3) in **Appendix B**. We will also present the J48 decision tree, the NBTree statistics, and the NaiveBayes statistics for the other data tables in **Appendix B**.

Table A1: Comparing the Accuracy of different Classification Algorithms for Diabetes Data 1 (See Table 4.8)

Algorithm	Algorithm Type	Prediction Accuracy
Zero R	Rule-based	50.79
JRIP	Rule-based	100
NNGE	Rule-based	100
J48	Decision Tree-based	100
NBTree	Decision Tree-based	100
NaiveBayes	Bayes Probabilistic Classifier	100

4.9.2 Association Rules

Association algorithms are a different type, as compared to classification algorithms, of data mining algorithms. Their purpose is not to predict the value of the class variable. Rather, the purpose of association algorithms is to learn rules, with a specified minimum confidence factor, that describe the relationship between attributes and other attributes, and between attributes and class. We will be using one association algorithm to learn association rules from each of our medical data tables. The association algorithm is called Apriori. To run the association algorithm, a prerequisite is that a discretize filter must be applied to all the attributes and class variables in the data. In other words, all the variables must be discrete not continuous. To apply the discretize filter, we select it as shown above and click Apply. We check to make sure the data is now completely discrete. Then, we can run the algorithm. For our example, to run the algorithm, we go to the Association tab. Then we select **choose**→**Apriori**, and then click **Start**.

For the *diabetes data 1* table 4.8, 103 rules with a confidence of 0.70 or higher are found by **Apriori**. The rules have the if then format: *if(condition & condition1 &) then result1 & result2 &* Below, we will selectively show only the rules where the result is a conclusion about the value of the primary diagnosis variable. For example, look at rule 7 below. It states that if the *fasting_plasma_glucose* is between $-\infty$ & 66.145 mg/dL then (\implies) the primary diagnosis is normal. The rule has a confidence of 1 (conf:(1)). Also, the minimum possible value for *fasting_plasma_glucose* is 0 mg/dL. So when the rule says if the *fasting_plasma_glucose* is between $-\infty$ & 66.145 mg/dL, it is equivalent to saying if the *fasting_plasma_glucose* is between 0 & 66.145 mg/dL. As another example, look at rule 48. It states if *gender* is male & *fasting_plasma_glucose* is between 150.134 & max, then the primary diagnosis is Diabetes Uncompl type II or Type 2 diabetes without complications.

- 7. *fasting_plasma_glucose*='(-inf-66.145453]' \implies *prim_diagnosis*=Normal conf:(1)
- 9. *ageDays*='(-inf-3.7]' *fasting_plasma_glucose*='(-inf-66.145453]' \implies *prim_diagnosis*=Normal conf:(1)
- 12. *fasting_plasma_glucose*='(150.13361-inf)' 12 \implies *prim_diagnosis*=DIABETES UNCOMPL TYPE II conf:(1)
- 14. *ageDays*='(-inf-3.7]' *fasting_plasma_glucose*='(150.13361-inf)' \implies *prim_diagnosis*=DIABETES UNCOMPL TYPE II conf:(1)
- 15. *fasting_plasma_glucose*='(150.13361-inf)' \implies *ageDays*='(-inf-3.7]' *prim_diagnosis*=DIABETES UNCOMPL TYPE II conf:(1)
- 17. *fasting_plasma_glucose*='(66.145453-76.643973]' \implies *prim_diagnosis*=Normal conf:(1)
- 21. *ageDays*='(-inf-3.7]' *gender*=1.0 *fasting_plasma_glucose*='(-inf-66.145453]' \implies *prim_diagnosis*=Normal conf:(1)
- 24. *fasting_plasma_glucose*='(139.635091-150.13361]' \implies *prim_diagnosis*=DIABETES UNCOMPL TYPE II conf:(1)
- 28. *ageDays*='(-inf-3.7]' *fasting_plasma_glucose*='(139.635091-150.13361]' \implies *prim_diagnosis*=DIABETES UNCOMPL TYPE II conf:(1)
- 29. *fasting_plasma_glucose*='(139.635091-150.13361]' \implies *ageDays*='(-inf-3.7]' *prim_diagnosis*=DIABETES UNCOMPL TYPE II conf:(1)

35. ageDays='(-inf-3.7]' fasting_plasma_glucose='(129.136571-139.635091]' ==> prim_diagnosis=DIABETES UNCOMPL TYPE II conf:(1)
36. fasting_plasma_glucose='(129.136571-139.635091]' ==> ageDays='(-inf-3.7]' prim_diagnosis=DIABETES UNCOMPL TYPE II conf:(1)
42. ageYears='(9.5-19]' fasting_plasma_glucose='(150.13361-inf)' ==> prim_diagnosis=DIABETES UNCOMPL TYPE II conf:(1)
46. ageDays='(-inf-3.7]' fasting_plasma_glucose='(76.643973-87.142493]' ==> prim_diagnosis=Normal conf:(1)
47. gender=0.0 fasting_plasma_glucose='(139.635091-150.13361]' ==> prim_diagnosis=DIABETES UNCOMPL TYPE II conf:(1)
48. gender=0.0 fasting_plasma_glucose='(150.13361-inf)' ==> prim_diagnosis=DIABETES UNCOMPL TYPE II conf:(1)
49. gender=1.0 fasting_plasma_glucose='(150.13361-inf)' ==> prim_diagnosis=DIABETES UNCOMPL TYPE II conf:(1)
51. ageYears='(9.5-19]' ageDays='(-inf-3.7]' fasting_plasma_glucose='(150.13361-inf)' ==> prim_diagnosis=DIABETES UNCOMPL TYPE II conf:(1)
- 57.ageDays='(-inf-3.7]' gender=0.0 fasting_plasma_glucose='(150.13361-inf)' ==> prim_diagnosis=DIABETES UNCOMPL TYPE II conf:(1)
- 60.ageDays='(-inf-3.7]' gender=1.0 fasting_plasma_glucose='(150.13361-inf)' ==> prim_diagnosis=DIABETES UNCOMPL TYPE II conf:(1)
74. ageYears='(-inf-9.5]' ageDays='(-inf-3.7]' gender=1.0 ==> prim_diagnosis=Normal conf:(0.88)
79. ageYears='(-inf-9.5]' ==> prim_diagnosis=Normal conf:(0.83)
82. ageYears='(-inf-9.5]' ageDays='(-inf-3.7]' ==> prim_diagnosis=Normal conf:(0.8)
84. ageYears='(9.5-19]' gender=0.0 ==> prim_diagnosis=DIABETES UNCOMPL TYPE II conf:(0.78)
86. ageYears='(9.5-19]' ageDays='(-inf-3.7]' gender=0.0 ==> prim_diagnosis=DIABETES UNCOMPL TYPE II conf:(0.78)

The rules that make a conclusion about the value of primary diagnosis for the rest of the data tables are presented in **Appendix C**.

Appendix B

Diabetes Data 2 – Classification Algorithms' Prediction Accuracies & Results

Table B1: Comparing the Accuracy of different Classification Algorithms for Diabetes Data 2 (See Table 4.10)

Algorithm	Algorithm Type	Prediction Accuracy
Zero R	Rule-based	65.10
JRIP	Rule-based	79.30
NNGE	Rule-based	100
J48	Decision Tree-based	84.11
NBTree	Decision Tree-based	78.26
NaiveBayes	Bayes Probabilistic Classifier	76.30

J48 Tree

Tree View

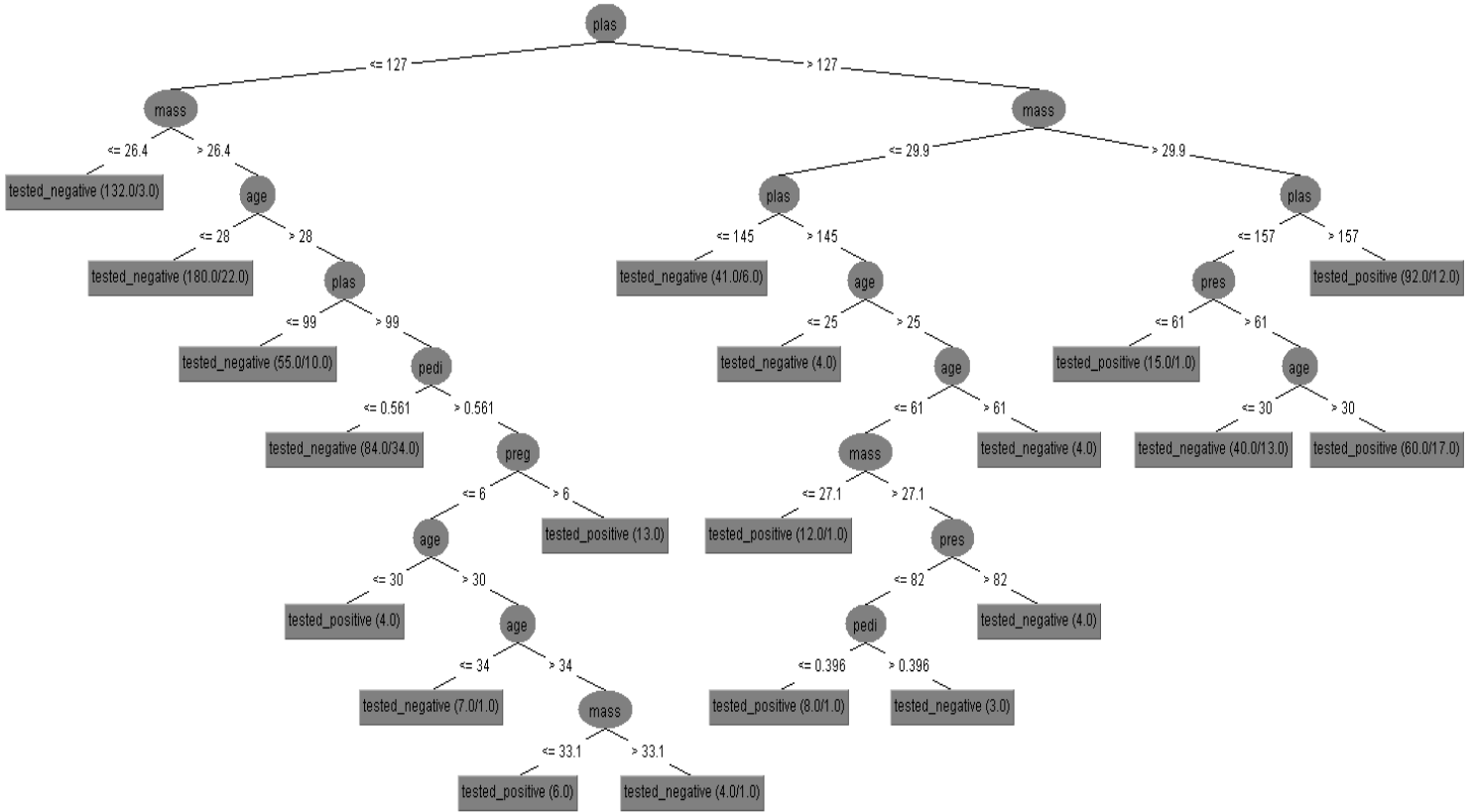


Fig. B1: J48 Decision Tree for Diabetes Data 2

NBTree

NBTree

: NB0

Leaf number: 0 Naive Bayes Classifier

	Class	
Attribute	tested_negative (0.65)	tested_positive (0.35)
preg		

'(-inf-6.5]'	427.0	174.0
'(6.5-inf)'	75.0	96.0
[total]	502.0	270.0
plas		
'(-inf-99.5]'	182.0	17.0
'(99.5-127.5]'	211.0	79.0
'(127.5-154.5]'	86.0	77.0
'(154.5-inf)'	25.0	99.0
[total]	504.0	272.0
pres		
'All'	501.0	269.0
[total]	501.0	269.0
skin		
'All'	501.0	269.0
[total]	501.0	269.0
insu		
'(-inf-14.5]'	237.0	140.0
'(14.5-121]'	165.0	28.0
'(121-inf)'	101.0	103.0
[total]	503.0	271.0
mass		
'(-inf-27.85]'	196.0	28.0
'(27.85-inf)'	306.0	242.0
[total]	502.0	270.0
pedi		
'(-inf-0.5275]'	362.0	149.0
'(0.5275-inf)'	140.0	121.0
[total]	502.0	270.0
age		
'(-inf-28.5]'	297.0	72.0
'(28.5-inf)'	205.0	198.0
[total]	502.0	270.0

Naive Bayes Classifier

Attribute	Class	
	tested_negative (0.65)	tested_positive (0.35)
=====		
preg		
mean	3.4234	4.9795
std. dev.	3.0166	3.6827
weight sum	500	268
precision	1.0625	1.0625
plas		
mean	109.9541	141.2581
std. dev.	26.1114	31.8728
weight sum	500	268
precision	1.4741	1.4741
pres		
mean	68.1397	70.718
std. dev.	17.9834	21.4094
weight sum	500	268
precision	2.6522	2.6522
skin		
mean	19.8356	22.2824
std. dev.	14.8974	17.6992
weight sum	500	268
precision	1.98	1.98
insu		
mean	68.8507	100.2812
std. dev.	98.828	138.4883
weight sum	500	268
precision	4.573	4.573
mass		
mean	30.3009	35.1475
std. dev.	7.6833	7.2537
weight sum	500	268

precision	0.2717	0.2717
pedi		
mean	0.4297	0.5504
std. dev.	0.2986	0.3715
weight sum	500	268
precision	0.0045	0.0045
age		
mean	31.2494	37.0808
std. dev.	11.6059	10.9146
weight sum	500	268
precision	1.1765	1.1765

Anemia Data – Classification Algorithms’ Prediction Accuracies & Results

Table B2: Comparing the Accuracy of different Classification Algorithms for Anemia Data(See Table 4.11)

Algorithm	Algorithm Type	Prediction Accuracy
Zero R	Rule-based	50.43
JRIP	Rule-based	99.15
NNGE	Rule-based	100
J48	Decision Tree-based	100
NBTree	Decision Tree-based	100
NaiveBayes	Bayes Probabilistic Classifier	96.58

J48 Tree

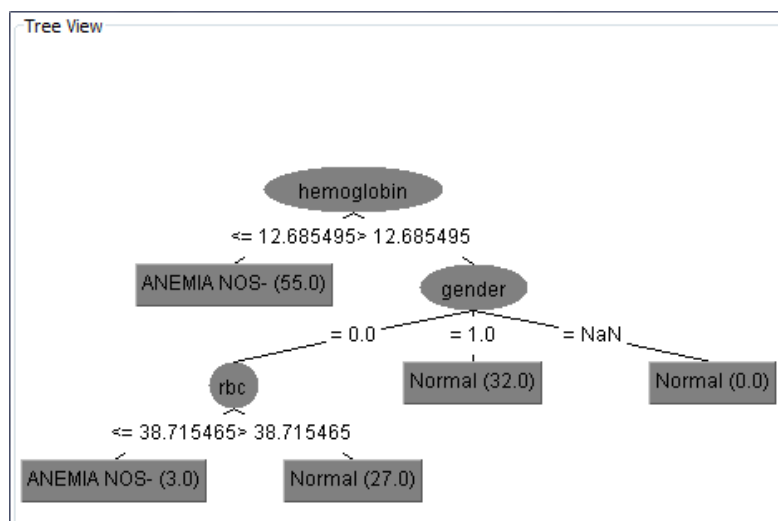


Fig. B2: J48 Decision Tree for Anemia Data

JRIP rules

(hemoglobin <= 12.685495) => prim_diagnosis=ANEMIA NOS- (55.0/0.0)
 (hemoglobin <= 13.415326) and (gender = 0.0) => prim_diagnosis=ANEMIA NOS-
 (2.0/0.0)
 => prim_diagnosis=Normal (60.0/1.0)

Naive Bayes Classifier

Attribute	Class	
	ANEMIA NOS- (0.5)	Normal (0.5)
=====		
ageYears		
mean	64.3765	56.931
std. dev.	25.6589	26.1904
weight sum	58	59
precision	1.6964	1.6964
ageDays		
mean	3	6.8814
std. dev.	12.8452	40.0767
weight sum	58	59
precision	58	58
gender		
0.0	21.0	28.0
1.0	39.0	33.0
NaN	1.0	1.0
[total]	61.0	62.0
rbc		
mean	31.3905	41.6596
std. dev.	3.419	3.7279
weight sum	58	59
precision	0.211	0.211
hemoglobin		

mean	10.1618	14.5302
std. dev.	1.3376	1.1866
weight sum	58	59
precision	0.0818	0.0818

NBTree

NBTree

gender = 0.0: NB 1
gender = 1.0: NB 2
gender = NaN: NB 3

Leaf number: 1 Naive Bayes Classifier

Attribute	Class	
	ANEMIA NOS- (0.43)	Normal (0.57)
=====		
ageYears		
'All'	21.0	28.0
[total]	21.0	28.0
ageDays		
'All'	21.0	28.0
[total]	21.0	28.0
gender		
0.0	21.0	28.0
1.0	1.0	1.0
NaN	1.0	1.0
[total]	23.0	30.0
rbc		
'(-inf-38.944935]'	21.0	1.0
'(38.944935-inf)'	1.0	28.0
[total]	22.0	29.0
hemoglobin		
'(-inf-13.524468]'	21.0	1.0

'(13.524468-inf)'	1.0	28.0
[total]	22.0	29.0

Leaf number: 2 Naive Bayes Classifier

Attribute	Class	
	ANEMIA NOS- (0.54)	Normal (0.46)
=====		
ageYears		
'All'	39.0	33.0
[total]	39.0	33.0
ageDays		
'All'	39.0	33.0
[total]	39.0	33.0
gender		
0.0	1.0	1.0
1.0	39.0	33.0
NaN	1.0	1.0
[total]	41.0	35.0
rbc		
'(-inf-34.829141]'	39.0	1.0
'(34.829141-inf)'	1.0	33.0
[total]	40.0	34.0
hemoglobin		
'(-inf-12.374101]'	39.0	1.0
'(12.374101-inf)'	1.0	33.0
[total]	40.0	34.0

Leaf number: 3 Naive Bayes Classifier

Attribute	Class	
	ANEMIA NOS- (0.5)	Normal (0.5)
=====		

ageYears		
'All'	1.0	1.0
[total]	1.0	1.0
ageDays		
'All'	1.0	1.0
[total]	1.0	1.0
gender		
0.0	1.0	1.0
1.0	1.0	1.0
NaN	1.0	1.0
[total]	3.0	3.0
rbc		
'All'	1.0	1.0
[total]	1.0	1.0
hemoglobin		
'All'	1.0	1.0
[total]	1.0	1.0

Blood Pressure Data – Classification Algorithms’ Prediction Accuracies & Results

Table B3: Comparing the Accuracy of different Classification Algorithms for Blood Pressure Data (See Table 4.12)

Algorithm	Algorithm Type	Prediction Accuracy
Zero R	Rule-based	50.36
JRIP	Rule-based	100
NNGE	Rule-based	100
J48	Decision Tree-based	100
NBTree	Decision Tree-based	100
NaiveBayes	Bayes Probabilistic Classifier	100

J48 Tree

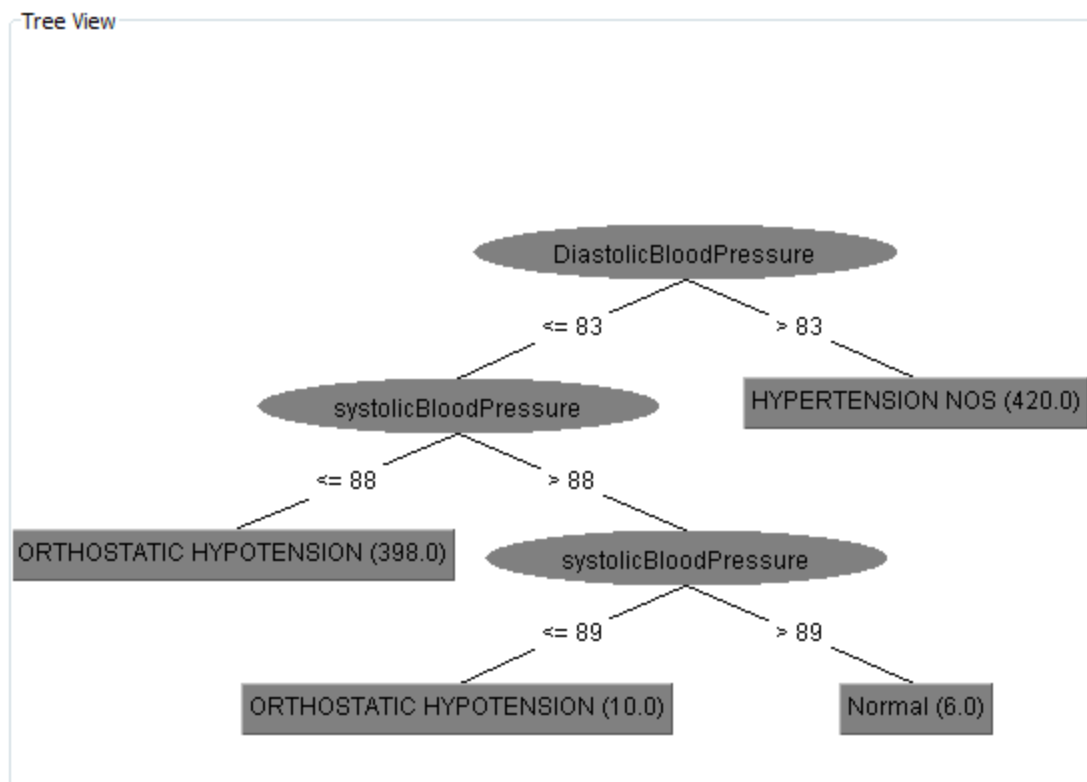


Fig. B3: J48 Decision Tree for Blood Pressure Data 1

JRIP

(ageYears <= 47) and (systolicBloodPressure <= 114) and
(systolicBloodPressure >= 98) => prim_diagnosis=Normal (6.0/0.0)
(systolicBloodPressure <= 89) => prim_diagnosis=ORTHOSTATIC
HYPOTENSION (408.0/0.0)
=> prim_diagnosis=HYPERTENSION NOS (420.0/0.0)

Naive Bayes

Naive Bayes Classifier

Attribute	Class		
	ORTHOSTATIC HYPOTENSION (0.49)	HYPERTENSION NOS (0.5)	Normal (0.01)
=====			
ageYears			
mean	72.3405	61.5066	33.8713
std. dev.	16.6323	21.2196	7.2291
weight sum	408	420	6
precision	1.2025	1.2025	1.2025
ageDays			
mean	0	0	0
std. dev.	0.0017	0.0017	0.0017
weight sum	408	420	6
precision	0.01	0.01	0.01
gender			
0.0	173.0	139.0	4.0
1.0	237.0	283.0	4.0
NaN	1.0	1.0	1.0
[total]	411.0	423.0	9.0
systolicBloodPressure			
mean	74.6272	153.3586	106.2051
std. dev.	8.7207	8.787	6.0203
weight sum	408	420	6
precision	1.6769	1.6769	1.6769
DiastolicBloodPressure			
mean	49.5123	104.2834	75.1697
std. dev.	5.7233	8.8772	7.0204
weight sum	408	420	6
precision	1.4364	1.4364	1.4364

NBTree

NBTree

: NBO

Leaf number: 0 Naive Bayes Classifier

Attribute	Class		
	ORTHOSTATIC HYPOTENSION (0.49)	HYPERTENSION NOS (0.5)	Normal (0.01)
=====			
ageYears			
'(-inf-62.5]'	87.0	227.0	7.0
'(62.5-inf)'	323.0	195.0	1.0
[total]	410.0	422.0	8.0
ageDays			
'All'	409.0	421.0	7.0
[total]	409.0	421.0	7.0
gender			
0.0	173.0	139.0	4.0
1.0	237.0	283.0	4.0
NaN	1.0	1.0	1.0
[total]	411.0	423.0	9.0
systolicBloodPressure			
'(-inf-93.5]'	409.0	1.0	1.0
'(93.5-127]'	1.0	1.0	7.0
'(127-inf)'	1.0	421.0	1.0
[total]	411.0	423.0	9.0
DiastolicBloodPressure			
'(-inf-61.5]'	409.0	1.0	1.0
'(61.5-86.5]'	1.0	1.0	7.0
'(86.5-inf)'	1.0	421.0	1.0
[total]	411.0	423.0	9.0

Calcemia Data – Classification Algorithms' Prediction Accuracies & Results

Algorithm	Algorithm Type	Prediction Accuracy
Zero R	Rule-based	71.43
JRIP	Rule-based	80.95
NNGE	Rule-based	100
J48	Decision Tree-based	100
NBTree	Decision Tree-based	100
NaiveBayes	Bayes Probabilistic Classifier	100

Table B4: Comparing the Accuracy of different Classification Algorithms for Calcemia (See Table 4.13)

J48 Tree

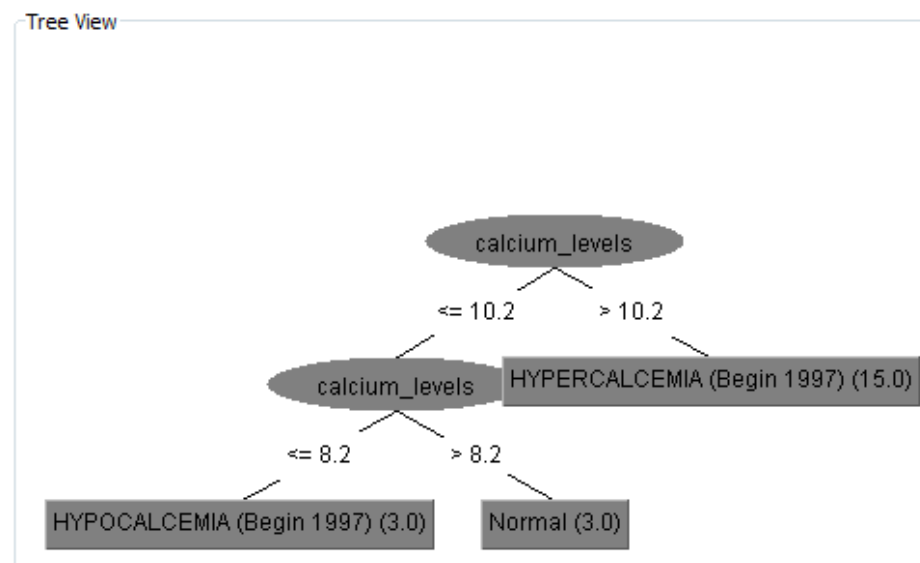


Fig. B4: J48 Decision Tree for Calcemia Data

Naive Bayes

Naive Bayes Classifier

Attribute	Class		
	HYPERCALCEMIA (Begin 1997) (0.67)	HYPOCALCEMIA (Begin 1997) (0.17)	Normal (0.17)
=====			
ageYears			
mean	69.1765	38.8235	37.0588
std. dev.	20.5919	8.9983	11.4366
weight sum	15	3	3
precision	5.2941	5.2941	5.2941
ageDays			
mean	2.9333	0	0
std. dev.	10.9755	7.3333	7.3333
weight sum	15	3	3
precision	44	44	44
gender			
0.0	7.0	2.0	2.0
1.0	10.0	3.0	3.0
NaN	1.0	1.0	1.0
[total]	18.0	6.0	6.0
calcium_levels			
mean	11.9052	7.9852	9.7481
std. dev.	0.8612	0.1467	0.388
weight sum	15	3	3
precision	0.3111	0.3111	0.3111

NBTree

=== Classifier model (full training set) ===

NBTree

: NBO

Leaf number: 0 Naive Bayes Classifier

Attribute	Class		
	HYPERCALCEMIA (Begin 1997) (0.67)	HYPOCALCEMIA (Begin 1997) (0.17)	Normal (0.17)
=====			
ageYears			
'(-inf-56.5]'	3.0	4.0	4.0
'(56.5-inf)'	14.0	1.0	1.0
[total]	17.0	5.0	5.0
ageDays			
'All'	16.0	4.0	4.0
[total]	16.0	4.0	4.0
gender			
0.0	7.0	2.0	2.0
1.0	10.0	3.0	3.0
NaN	1.0	1.0	1.0
[total]	18.0	6.0	6.0
calcium_levels			
'(-inf-8.7]'	1.0	4.0	1.0
'(8.7-10.45]'	1.0	1.0	4.0
'(10.45-inf)'	16.0	1.0	1.0
[total]	18.0	6.0	6.0

JRIP

JRIP rules:

=====

(calcium_levels <= 8) => prim_diagnosis=HYPOCALCEMIA (Begin 1997) (2.0/0.0)
=> prim_diagnosis=HYPERCALCEMIA (Begin 1997) (19.0/4.0)

Appendix C

Selected Association Rules for Diabetes Data 2 (Table 4.9)

2. skin='(-inf-9.9]' pedi='(-inf-0.3122]' class=tested_negative ==> insu='(-inf-84.6]' conf:(1)
7. mass='(20.13-26.84]' age='(-inf-27]' ==> class=tested_negative conf:(0.97)
8. plas='(79.6-99.5]' age='(-inf-27]' ==> class=tested_negative conf:(0.95)
9. plas='(79.6-99.5]' insu='(-inf-84.6]' ==> class=tested_negative conf:(0.94)
10. skin='(19.8-29.7]' age='(-inf-27]' ==> class=tested_negative conf:(0.93)
11. plas='(79.6-99.5]' ==> class=tested_negative conf:(0.92)
12. insu='(-inf-84.6]' mass='(20.13-26.84]' ==> class=tested_negative conf:(0.91)
13. mass='(20.13-26.84]' ==> class=tested_negative conf:(0.9)
14. plas='(99.5-119.4]' age='(-inf-27]' ==> class=tested_negative conf:(0.89)
15. insu='(-inf-84.6]' pedi='(-inf-0.3122]' age='(-inf-27]' ==> class=tested_negative conf:(0.89)
16. pedi='(-inf-0.3122]' age='(-inf-27]' ==> class=tested_negative conf:(0.88)
17. skin='(9.9-19.8]' ==> class=tested_negative conf:(0.87)
18. pres='(61-73.2]' age='(-inf-27]' ==> class=tested_negative conf:(0.85)
9. mass='(26.84-33.55]' age='(-inf-27]' ==> class=tested_negative conf:(0.85)
20. preg='(-inf-1.7]' insu='(-inf-84.6]' age='(-inf-27]' ==> class=tested_negative conf:(0.84)
21. insu='(-inf-84.6]' age='(-inf-27]' ==> class=tested_negative conf:(0.83)
22. preg='(-inf-1.7]' age='(-inf-27]' ==> class=tested_negative conf:(0.83)
23. preg='(1.7-3.4]' insu='(-inf-84.6]' ==> class=tested_negative conf:(0.83)
25. preg='(1.7-3.4]' age='(-inf-27]' ==> class=tested_negative conf:(0.82)
26. age='(-inf-27]' ==> class=tested_negative conf:(0.82)
27. pres='(48.8-61]' ==> class=tested_negative conf:(0.81)
29. plas='(99.5-119.4]' pedi='(-inf-0.3122]' ==> class=tested_negative conf:(0.81)
30. pedi='(0.3122-0.5464]' age='(-inf-27]' ==> class=tested_negative conf:(0.8)
31. preg='(-inf-1.7]' class=tested_negative ==> age='(-inf-27]' conf:(0.8)
32. preg='(-inf-1.7]' insu='(-inf-84.6]' ==> class=tested_negative conf:(0.78)
35. plas='(79.6-99.5]' ==> insu='(-inf-84.6]' class=tested_negative conf:(0.76)
36. insu='(-inf-84.6]' pedi='(-inf-0.3122]' ==> class=tested_negative conf:(0.75)
38. pres='(73.2-85.4]' pedi='(-inf-0.3122]' ==> class=tested_negative conf:(0.75)

Selected Association Rules for Blood Pressure (Table 4.10)

11. systolicBloodPressure='(147.2-158.1]' ==> prim_diagnosis=HYPERTENSION NOS conf:(1)
12. DiastolicBloodPressure='(47.9-55.8]' ==> prim_diagnosis=ORTHOSTATIC HYPOTENSION conf:(1)
14. ageDays='All' systolicBloodPressure='(147.2-158.1]' ==> prim_diagnosis=HYPERTENSION NOS conf:(1)

15. systolicBloodPressure='(147.2-158.1]' ==> ageDays='All' prim_diagnosis=HYPERTENSION NOS conf:(1)

17. ageDays='All' DiastolicBloodPressure='(47.9-55.8]' ==> prim_diagnosis=ORTHOSTATIC HYPOTENSION conf:(1)

20. DiastolicBloodPressure='(-inf-47.9]' ==> prim_diagnosis=ORTHOSTATIC HYPOTENSION conf:(1)

22. ageDays='All' DiastolicBloodPressure='(-inf-47.9]' ==> prim_diagnosis=ORTHOSTATIC HYPOTENSION conf:(1)

26. systolicBloodPressure='(-inf-70.9]' ==> prim_diagnosis=ORTHOSTATIC HYPOTENSION conf:(1)

29. ageDays='All' systolicBloodPressure='(-inf-70.9]' ==> prim_diagnosis=ORTHOSTATIC HYPOTENSION conf:(1)

34. systolicBloodPressure='(70.9-81.8]' ==> prim_diagnosis=ORTHOSTATIC HYPOTENSION conf:(1)

36. ageDays='All' systolicBloodPressure='(70.9-81.8]' ==> prim_diagnosis=ORTHOSTATIC HYPOTENSION conf:(1)

41. ageDays='All' systolicBloodPressure='(158.1-inf)' ==> prim_diagnosis=HYPERTENSION NOS conf:(1)

46. ageDays='All' systolicBloodPressure='(136.3-147.2]' 125 ==> prim_diagnosis=HYPERTENSION NOS 125 conf:(1)

49. systolicBloodPressure='(81.8-92.7]' 121 ==> prim_diagnosis=ORTHOSTATIC HYPOTENSION 121 conf:(1)

51. ageDays='All' systolicBloodPressure='(81.8-92.7]' 121 ==> prim_diagnosis=ORTHOSTATIC HYPOTENSION 121 conf:(1)

55. gender=1.0 systolicBloodPressure='(147.2-158.1]' 117 ==> prim_diagnosis=HYPERTENSION NOS 117 conf:(1)

57. ageDays='All' gender=1.0 systolicBloodPressure='(147.2-158.1]' 117 ==> prim_diagnosis=HYPERTENSION NOS 117 conf:(1)

62. ageDays='All' DiastolicBloodPressure='(103.2-111.1]' 116 ==> prim_diagnosis=HYPERTENSION NOS 116 conf:(1)

67. ageDays='All' DiastolicBloodPressure='(111.1-inf)' 112 ==> prim_diagnosis=HYPERTENSION NOS 112 conf:(1)

71. DiastolicBloodPressure='(87.4-95.3]' 98 ==> prim_diagnosis=HYPERTENSION NOS 98 conf:(1)

74. ageDays='All' DiastolicBloodPressure='(87.4-95.3]' 98 ==> prim_diagnosis=HYPERTENSION NOS 98 conf:(1)

77. gender=1.0 DiastolicBloodPressure='(-inf-47.9]' 96 ==> prim_diagnosis=ORTHOSTATIC HYPOTENSION 96 conf:(1)

79. ageDays='All' gender=1.0 DiastolicBloodPressure='(-inf-47.9]' 96 ==> prim_diagnosis=ORTHOSTATIC HYPOTENSION 96 conf:(1)

84. ageDays='All' gender=1.0 DiastolicBloodPressure='(47.9-55.8]' 95 ==> prim_diagnosis=ORTHOSTATIC HYPOTENSION 95 conf:(1)

89. ageDays='All' DiastolicBloodPressure='(95.3-103.2]' 94 ==> prim_diagnosis=HYPERTENSION NOS 94 conf:(1)

94. gender=1.0 systolicBloodPressure='(-inf-70.9]' 88 ==> prim_diagnosis=ORTHOSTATIC HYPOTENSION 88 conf:(1)

96. ageDays='All' gender=1.0 systolicBloodPressure='(-inf-70.9]' 88 ==> prim_diagnosis=ORTHOSTATIC HYPOTENSION 88 conf:(1)

100. gender=1.0 systolicBloodPressure='(158.1-inf)' 86 ==> prim_diagnosis=HYPERTENSION NOS 86 conf:(1)

102. ageDays='All' gender=1.0 systolicBloodPressure='(158.1-inf)' 86 ==> prim_diagnosis=HYPERTENSION NOS 86 conf:(1)

Selected Association Rules for Calcemia (Table 4.11)

4. ageYears='(63-72]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

6. ageYears='(72-81]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

8. calcium_levels='(10.6-11.16]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

10. ageYears='(63-72]' ageDays='(-inf-4.4]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

14. ageYears='(72-81]' ==> ageDays='(-inf-4.4]' prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

16. ageDays='(-inf-4.4]' calcium_levels='(10.6-11.16]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

20. ageYears='(81-inf)' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

28. calcium_levels='(-inf-8.36]' ==> prim_diagnosis=HYPOCALCEMIA (Begin 1997) conf:(1)

29. calcium_levels='(11.72-12.28]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

30. calcium_levels='(12.28-12.84]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

31. calcium_levels='(12.84-inf)' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

33. ageYears='(72-81]' gender=1.0 ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

35. ageYears='(81-inf)' ageDays='(-inf-4.4]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

42. ageDays='(-inf-4.4]' calcium_levels='(-inf-8.36]' ==> prim_diagnosis=HYPOCALCEMIA (Begin 1997) conf:(1)

46. ageDays='(-inf-4.4]' calcium_levels='(12.28-12.84]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

49. ageDays='(-inf-4.4]' calcium_levels='(12.84-inf)' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

52. gender=1.0 calcium_levels='(12.28-12.84]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

55. ageYears='(72-81]' ageDays='(-inf-4.4]' gender=1.0 ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

59. ageDays='(-inf-4.4]' gender=1.0 calcium_levels='(12.28-12.84]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

3 conf:(1)

70. calcium_levels='(11.16-11.72]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

76. ageYears='(54-63]' ageDays='(-inf-4.4]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

79. ageYears='(54-63]' gender=1.0 ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

84. ageYears='(63-72]' gender=0.0 ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

87. ageYears='(63-72]' gender=1.0 ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

88. ageYears='(63-72]' calcium_levels='(12.28-12.84]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

90. ageYears='(72-81]' calcium_levels='(10.6-11.16]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

93. ageYears='(81-inf)' gender=1.0 ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

94. ageYears='(81-inf)' calcium_levels='(10.6-11.16]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

105. ageDays='(-inf-4.4]' calcium_levels='(11.16-11.72]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

107. ageDays='(-inf-4.4]' calcium_levels='(11.72-12.28]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

108. gender=0.0 calcium_levels='(10.6-11.16]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

109. gender=0.0 calcium_levels='(11.72-12.28]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

110. gender=0.0 calcium_levels='(12.84-inf)' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

112. gender=1.0 calcium_levels='(-inf-8.36]' ==> prim_diagnosis=HYPOCALCEMIA (Begin 1997) conf:(1)

113. gender=1.0 calcium_levels='(10.6-11.16]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

115. gender=1.0 calcium_levels='(11.16-11.72]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

119. ageYears='(54-63]' ageDays='(-inf-4.4]' gender=1.0 ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

125. ageYears='(63-72]' ageDays='(-inf-4.4]' gender=0.0 ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

133. ageYears='(63-72]' ageDays='(-inf-4.4]' gender=1.0 ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

136. ageYears='(63-72]' ageDays='(-inf-4.4]' calcium_levels='(12.28-12.84]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

140. ageYears='(63-72]' gender=1.0 calcium_levels='(12.28-12.84]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

144. ageYears='(72-81]' ageDays='(-inf-4.4]' calcium_levels='(10.6-11.16]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

147. ageYears='(81-inf)' ageDays='(-inf-4.4]' gender=1.0 ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

150. ageYears='(81-inf)' ageDays='(-inf-4.4]' calcium_levels='(10.6-11.16]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

153. ageDays='(-inf-4.4]' gender=0.0 calcium_levels='(10.6-11.16]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

156. ageDays='(-inf-4.4]' gender=0.0 calcium_levels='(12.84-inf)' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) 2 conf:(1)

160. ageDays='(-inf-4.4]' gender=1.0 calcium_levels='(-inf-8.36]' ==> prim_diagnosis=HYPOCALCEMIA (Begin 1997) 2 conf:(1)

164. ageDays='(-inf-4.4]' gender=1.0 calcium_levels='(10.6-11.16]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

168. ageDays='(-inf-4.4]' gender=1.0 calcium_levels='(11.16-11.72]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

171. ageDays='(-inf-4.4]' calcium_levels='(11.16-11.72]' ==> gender=1.0 prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

176. ageYears='(63-72]' ageDays='(-inf-4.4]' gender=1.0 calcium_levels='(12.28-12.84]' ==> prim_diagnosis=HYPERCALCEMIA (Begin 1997) conf:(1)

Selected Association Rules for Calcemia (Table 4.12)

5. hemoglobin='(9.691263-10.640166]' ==> prim_diagnosis=ANEMIA NOS- [conf:\(1\)](#)

9. hemoglobin='(14.435776-15.384679]' ==> prim_diagnosis=Normal [conf:\(1\)](#)

10. ageDays='(-inf-29]' hemoglobin='(9.691263-10.640166]' ==> prim_diagnosis=ANEMIA NOS- [conf:\(1\)](#)

13. ageDays='(-inf-29]' hemoglobin='(14.435776-15.384679]' ==> prim_diagnosis=Normal [conf:\(1\)](#)

18. rbc='(32.365973-34.813761]' ==> prim_diagnosis=ANEMIA NOS- [conf:\(1\)](#)

20. ageDays='(-inf-29]' rbc='(32.365973-34.813761]' ==> prim_diagnosis=ANEMIA NOS- [conf:\(1\)](#)

23. hemoglobin='(8.742361-9.691263]' ==> prim_diagnosis=ANEMIA NOS- [conf:\(1\)](#)

25. ageDays='(-inf-29]' hemoglobin='(8.742361-9.691263]' ==> prim_diagnosis=ANEMIA NOS- [conf:\(1\)](#)

27. rbc='(29.918184-32.365973]' ==> prim_diagnosis=ANEMIA NOS- [conf:\(1\)](#)

31. rbc='(39.709338-42.157127]' ==> prim_diagnosis=Normal [conf:\(1\)](#)

33. ageDays='(-inf-29]' rbc='(29.918184-32.365973]' ==> prim_diagnosis=ANEMIA NOS- [conf:\(1\)](#)

35. ageDays='(-inf-29]' rbc='(39.709338-42.157127]' ==> prim_diagnosis=Normal [conf:\(1\)](#)

39. rbc='(44.604915-47.052703]' ==> prim_diagnosis=Normal [conf:\(1\)](#)

46. ageDays='(-inf-29]' rbc='(44.604915-47.052703]' ==> prim_diagnosis=Normal [conf:\(1\)](#)
49. gender=0.0 rbc='(44.604915-47.052703]' ==> prim_diagnosis=Normal [conf:\(1\)](#)
51. gender=1.0 rbc='(32.365973-34.813761]' ==> prim_diagnosis=ANEMIA NOS- [conf:\(1\)](#)
53. gender=1.0 hemoglobin='(12.537971-13.486874]' ==> prim_diagnosis=Normal [conf:\(1\)](#)
56. ageDays='(-inf-29]' gender=0.0 rbc='(44.604915-47.052703]' ==> prim_diagnosis=Normal [conf:\(1\)](#)
62. ageDays='(-inf-29]' gender=1.0 rbc='(32.365973-34.813761]' ==> prim_diagnosis=ANEMIA NOS- [conf:\(1\)](#)
68. hemoglobin='(13.486874-14.435776]' ==> prim_diagnosis=Normal [conf:\(0.95\)](#)
71. ageDays='(-inf-29]' hemoglobin='(13.486874-14.435776]' ==> prim_diagnosis=Normal [conf:\(0.95\)](#)
86. hemoglobin='(12.537971-13.486874]' ==> prim_diagnosis=Normal [conf:\(0.8\)](#)
89. ageYears='(85.5-inf)' ageDays='(-inf-29]' ==> prim_diagnosis=ANEMIA NOS- [conf:\(0.76\)](#)

Appendix D

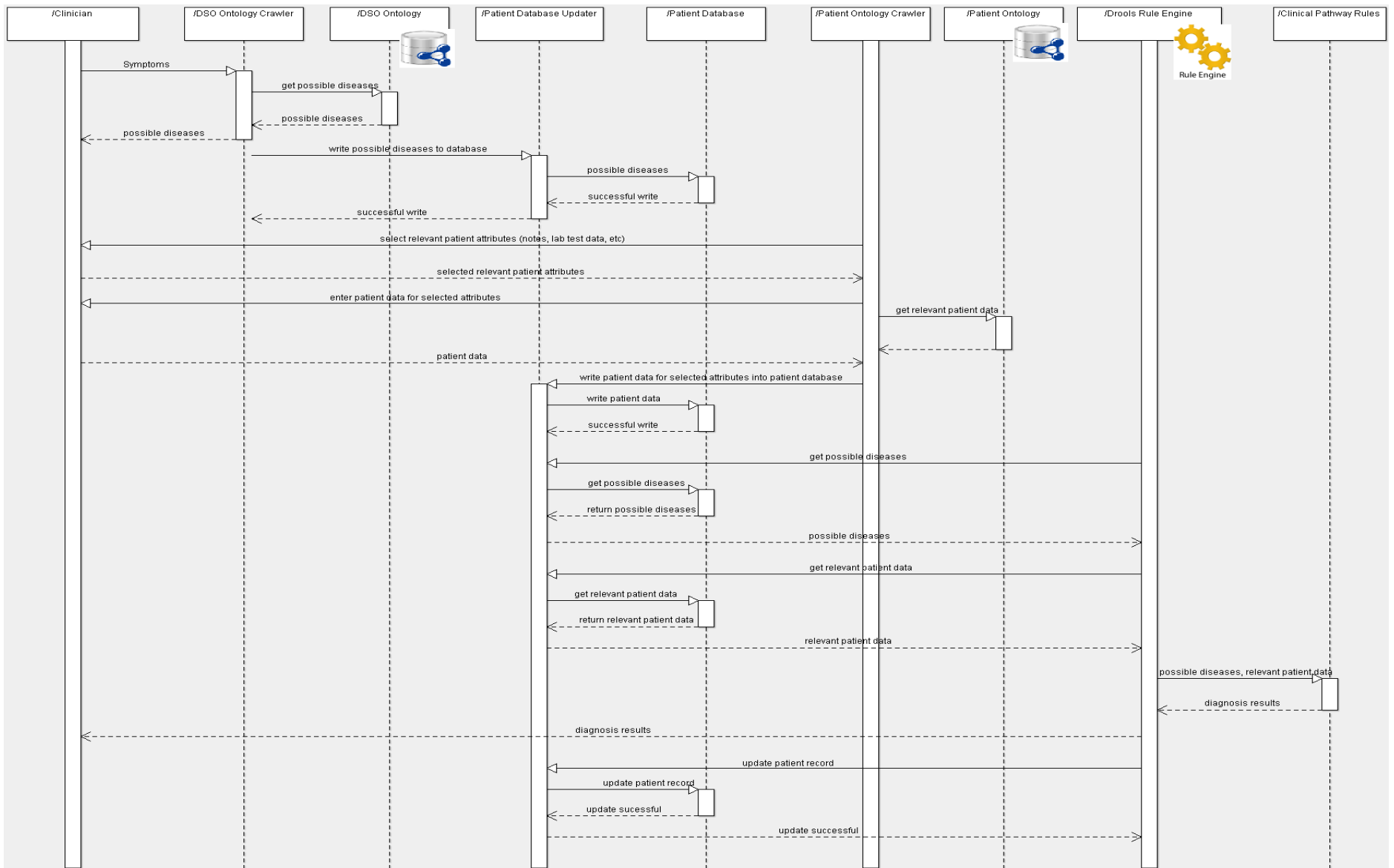


Fig. D1: Sequence Diagram for the Evidence-based DDX Recommender

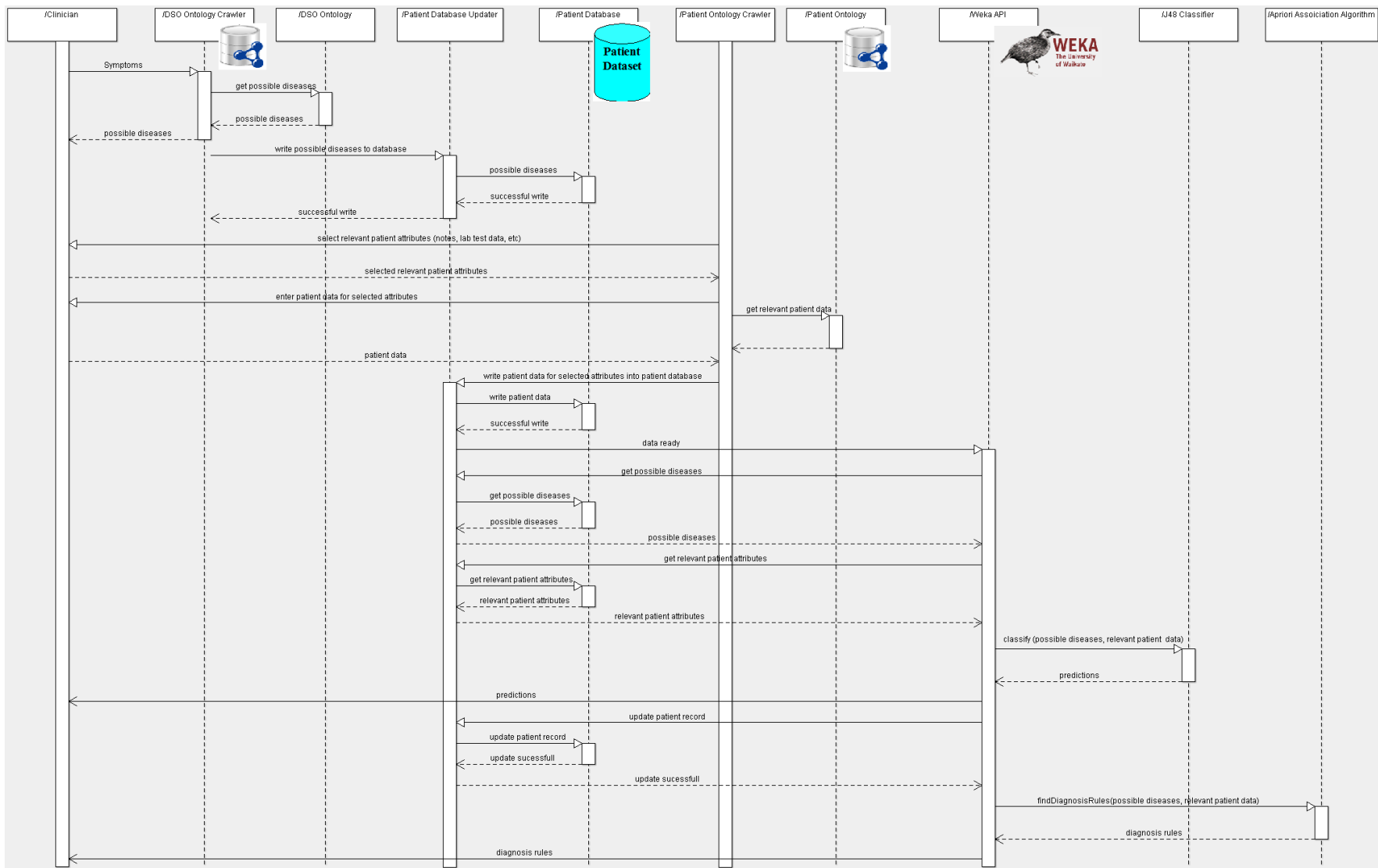


Fig. D2: Sequence Diagram for the Proximity-based DDX Recommender

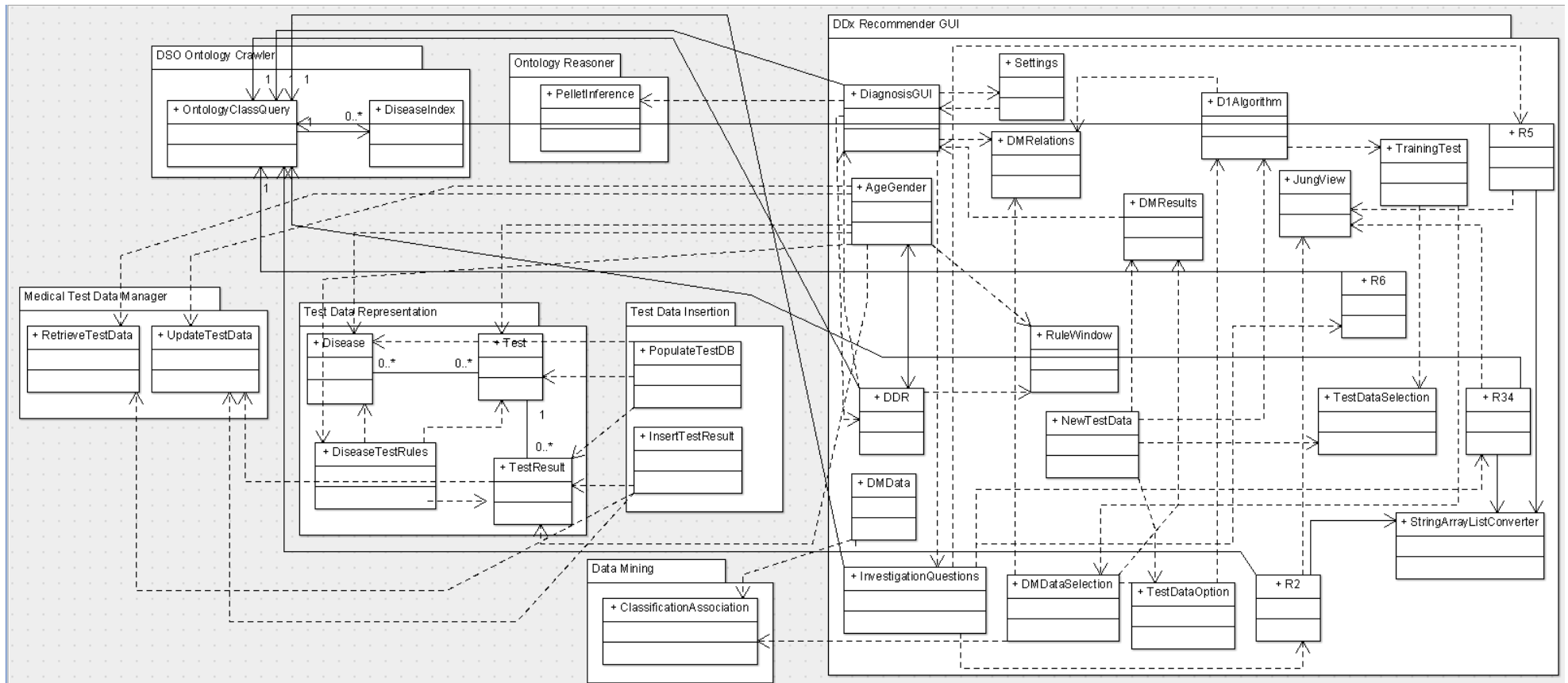


Fig. D3: Class Diagram for the Overall DDX Recommender

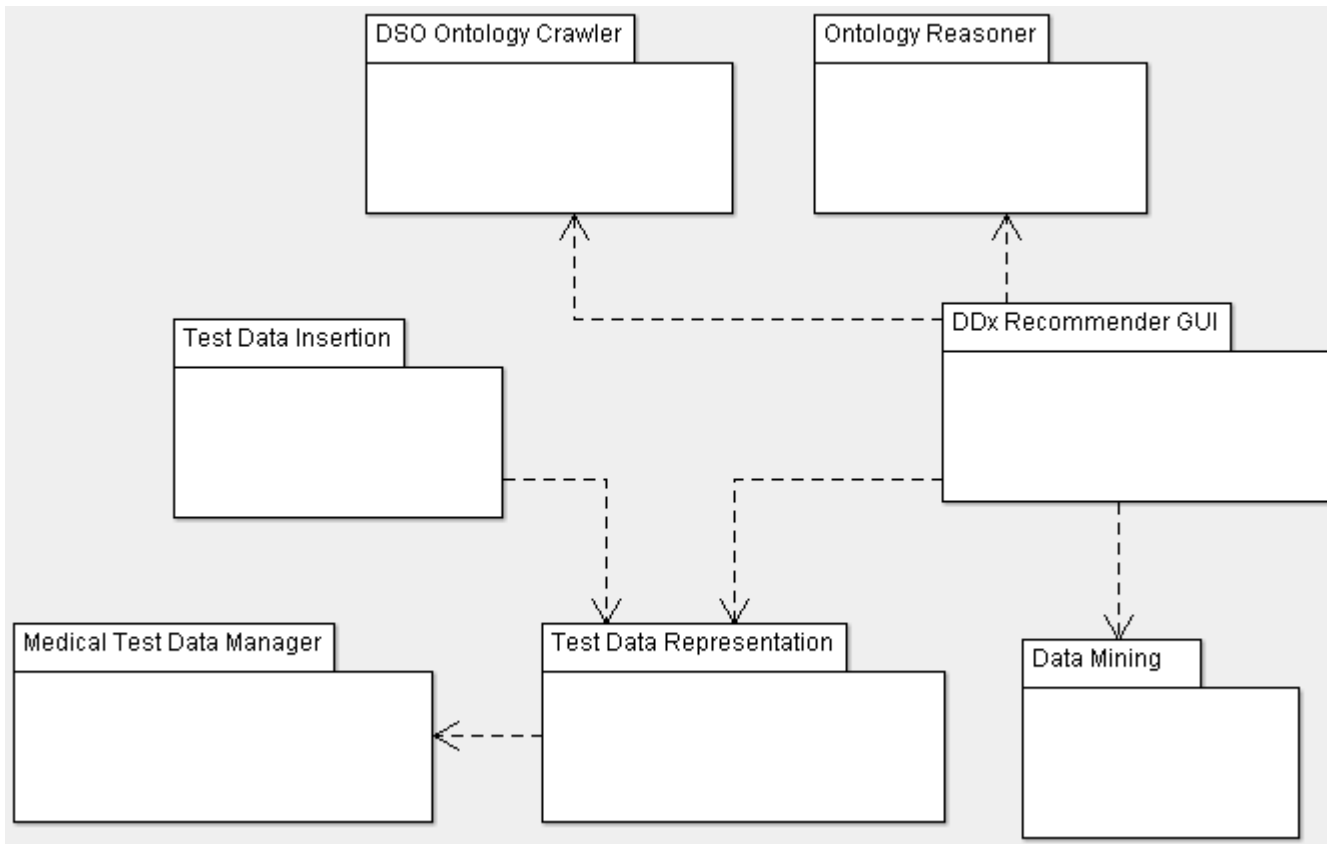


Fig. D4: Component Diagram of the Overall DDX Recommender

Appendix E

Table E1: Comparison of Equivalent Proximity and Evidence-based Rule Sets

Rule Condition's Attribute	Related Diagnosis	Evidence-based Rule	Proximity-based Rule	Number of Patient Records	Accuracy of Proximity-based Rule (%)
Maximum Normal FPG	Diabetes	109.8 mg/dL	76.6 mg/dL	64	69.8
Minimum Diabetic FPG	Diabetes	125 mg/dL	139.6 mg/dL	64	89.5
Minimum Hypertensive Systolic Blood Pressure	Hypertension	140 mmHg	147.2 mmHg	860	95.0
Minimum Normal Systolic Blood Pressure	Hypertension	90 mmHg	92.7 mmHg	860	97.0
Minimum Hypertensive Diastolic Blood Pressure	Hypertension	90 mmHg	87.4 mmHg	860	97.0
Minimum Normal Diastolic Blood Pressure	Hypertension	60 mmHg	55.8 mmHg	860	93.0
Normal Percentage of Red Blood Cells in the Blood for Males	Anemia	38.8 - 50.0 %	39.7 - 47.1 %	120	96.0
Normal Percentage of Red Blood Cells in the Blood for Females	Anemia	34.9 - 44.5 %	32.4 - 47.1 %	120	93.7
Normal Hemoglobin Levels in the Blood for Males	Anemia	13.5 - 17.5 mg/dL	12.5 - 15.4 mg/dL	120	90.3
Normal Hemoglobin	Anemia	12 - 15.5 mg/dL	12.5 - 15.4 mg/dL	120	97.7

Levels in the Blood for Females					
Maximum Hypotensive Diastolic Blood Pressure	Hypotension	60 mmHg	55.8 mmHg	860	93.0
Minimum Hypotensive Diastolic Blood Pressure	Hypotension	40 mmHg	47.9 mmHg	860	83.5
Minimum Bound for High Calcium Levels in the Blood	Calcemia	10.5 mg/dL	10.6 mg/dL	23	99.1
Minimum Normal of Calcium Levels in the Blood	Calcemia	9 mg/dL	9.48 mg/dL	23	94.9

Table E2: Accuracy of Proximity-based Rules as a function of Number of known Patient Cases available to Data Mining Algorithm

Number of Patient Records	Average Accuracy of Proximity-based Rule
64	79.7
860	95.5
120	94.4

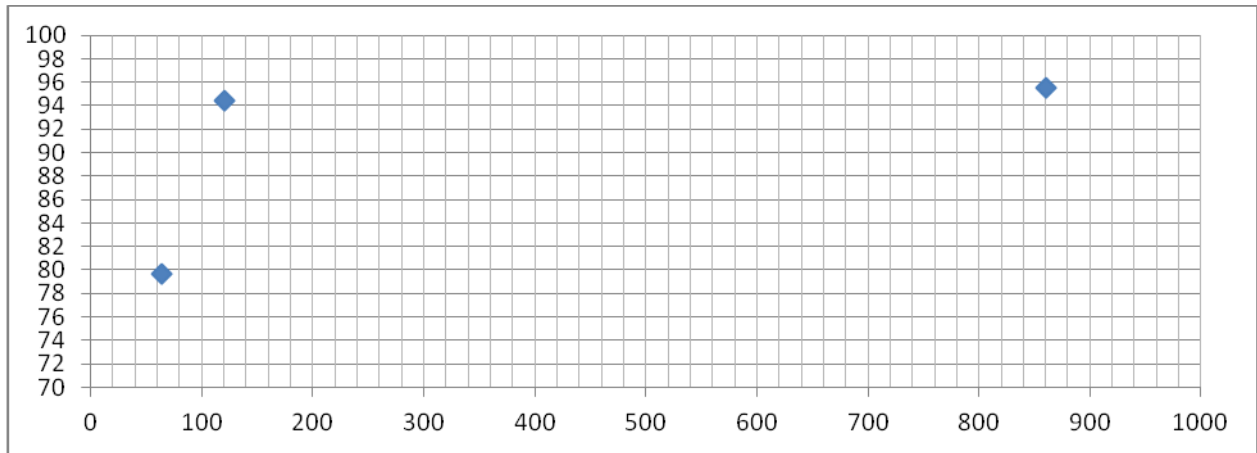


Figure E1: Graph of Accuracy of Proximity-based Rules as a function of Number of known Patient Cases available to Data Mining Algorithm

Table E3: Comparison of Proximity-based vs. Evidence-based Predictions for Anemia Patient Cases
(Accuracy of Proximity-based Recommender = $17/23 = 74\%$)

Age	Gender	RBC (%)	Hemoglobin Levels (mg/dL)	Evidence-based Diagnosis	Proximity-based Diagnosis
17	Female	33.77	10.11	Positive for Anemia	Positive for Anemia
95	Female	34.75	11.39	Positive for Anemia	Positive for Anemia
74	Male	33.39	9.30	Positive for Anemia	Positive for Anemia
50	Male	42.27	14.97	Normal	Normal
63	Male	41.66	13.83	Normal	Normal
82	Male	44.64	16.43	Normal	Normal
44	Male	42.67	16.10	Normal	Normal
59	Female	41.40	14.96	Normal	Normal
62	Female	33.18	10.36	Positive for Anemia	Positive for Anemia
19	Male	39.00	12.91	Positive for Anemia	Normal
26	Female	33.18	13.25	Positive for Anemia	Normal
54	Male	39.10	12.27	Positive for Anemia	Positive for Anemia
36	Female	40.12	12.27	Normal	Positive for Anemia
42	Female	33.18	12.27	Positive for Anemia	Positive for Anemia
56	Female	29.25	15.13	Positive for	Normal

				Anemia	
51	Female	29.25	11.43	Positive for Anemia	Positive for Anemia
41	Male	36.75	12.78	Positive for Anemia	Normal
36	Male	48.00	14.43	Normal	Normal
64	Male	43.67	16.23	Normal	Normal
46	Male	39.00	12.78	Positive for Anemia	Normal
68	Female	37.00	12.9	Normal	Normal
91	Female	32.95	11.86	Positive for Anemia	Positive for Anemia
59	Female	41.40	14.96	Normal	Normal

Table E4: Comparison of Proximity-based vs. Evidence-based Predictions for Diabetes Patient Cases
(Accuracy of Proximity-based Recommender = $3/4 = 75\%$)

Age	Gender	Fasting Plasma Glucose (mg/dL)	Evidence-based Diagnosis	Proximity-based Diagnosis
80	Female	136.28	Diabetes	Diabetes
95	Female	64.42	Normal	Normal
45	Female	101.60	Normal	Diabetes
38	Male	129.9	Diabetes	Diabetes