



Faculty of Sciences

Department of Computer Science and Information Systems

AN INTELLIGENT ELECTRONIC PRESCRIPTION ONTOLOGY FRAMEWORK FOR HEALTHCARE INTEROPERABILITY

by

OKON EUNICE CHINATU

Student ID number: 19100040

MSc (INFORMATION SYSTEMS AND DATA MANAGEMENT) (BIUST)

A Dissertation/Thesis Submitted to the Faculty of Sciences in Partial Fulfilment of the
Requirements for the Award of the Degree of Master of Science in Computer Science
and Information Systems of BIUST

Supervisor(s): Dr. Sigwele Tshiamo, Dr. Galani Malatsi

Department of Computer Science and Information Systems

Faculty of Science, BIUST

sigwelet@biust.ac.bw, galanim@biust.ac.bw

January, 2023

Table of Content

DECLARATION AND COPYRIGHT	V
CERTIFICATION	VI
ACKNOWLEDGEMENTS	VII
DEDICATION	VIII
ABSTRACT	IX
LIST OF FIGURES	X
LIST OF TABLES	XIII
ACRONYMS	XV
CHAPTER 1 INTRODUCTION	1
1.0 INTRODUCTION	1
1.1 BACKGROUND	1
1.2 GENERAL PROBLEM	2
1.3 PROBLEM STATEMENT	3
1.4 RESEARCH GOAL AND OBJECTIVES	3
1.4.1 Main Goal	3
1.4.2 Objectives	3
1.5 RESEARCH QUESTIONS	4
1.6 RESEARCH CONTRIBUTION	5
1.6.1 Research Papers	5
1.6.2 Research Work Contributions	5
1.7 SIGNIFICANCE OF STUDY	6
1.8 RELEVANCE TO THE DISCIPLINE	6
1.9 EVIDENCE OF ORIGINALITY	7
1.10 ORGANIZATION OF THE DISSERTATION	7
CHAPTER 2 LITERATURE REVIEW	9
2.0 INTRODUCTION	9
2.1 HEALTHCARE INTEROPERABILITY	9
2.2 TYPES/LEVELS OF HEALTHCARE INTEROPERABILITY	11
2.2.1 Foundational interoperability:	11
2.2.2 Structural Interoperability:	11

2.2.3 Semantic Interoperability:.....	11
2.2.4 Organizational Interoperability.....	12
2.3 INTEROPERABILITY APPROACHES:	13
2.4 INTEROPERABILITY STANDARDS.....	14
2.5 ONTOLOGY	16
2.5.1 Comparison Between Ontologies and Databases	16
2.5.2 Ontology Development Methodologies	17
2.5.3 Summary of Ontology Development Methodology	18
2.5.4 Ontology Languages	19
2.5.5 Summary of Ontology Language	20
2.5.6 Ontology Development Tools.....	21
2.5.7 Summary of Ontology Development Tools	23
2.5.8 Ontology Evaluation Tools	23
2.6 HEALTHCARE INTEROPERABILITY USING SEMANTIC E-PRESCRIPTION.....	25
2.6.1 E-Prescription Review.....	25
2.6.2 A review on Healthcare Interoperability ePrescription Frameworks	27
2.6.3 Summary of eprescription frameworks review	30
CHAPTER 3 RESEARCH METHODOLOGY: PROPOSED IEPO FRAMEWORK	32
3.0 INTRODUCTION.....	32
3.1 ONTOLOGY DEVELOPMENT FOR THE PROPOSED IEPO FRAMEWORK.....	38
3.2 CHAPTER TERMINOLOGIES.	39
3.3 ACQUISITION ACTIVITY	39
3.4 SPECIFICATION ACTIVITY.....	40
3.5 CONCEPTUALIZATION.....	41
3.5.1 Building the Glossary of Terms	42
3.5.2 Build the concept-classification trees:.....	49
3.5.3 Binary Relations Diagram	50
3.5.4 Concept Dictionary.....	51
3.5.5 A Binary-Relation Table	53
3.5.6 The Instance Attribute Table.....	55
3.5.7 Class Attribute Table	59
3.5.8 Formula Table	60
3.5.9 Constant Table	61
3.5.10 Formal Axioms Table of the Clinic and Pharmacy Ontology	61

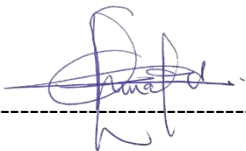
3.5.11 An Instance Table.....	63
3.6 INTERGRATION: ONTOLOGY MERGING	64
3.7 FORMALIZATION	65
3.7.1 Class and subclass creation in Protégé	67
3.7.2 IEPO Object Properties Creation in Protégé.....	68
3.7.3 IEPO Data Properties Creation in Protégé.....	68
3.7.4 Linking IEPO Classes Using Object Properties	69
3.7.5 Creation of IEPO Individuals or Instances: Automatic Individual Import with Cellfie Plugin.....	70
3.7.6 Linking IEPO Individuals Using Object Properties via Cellfie Plugin	72
3.7.7 Creating Instances of Data Properties (Attributes) for Imported Individuals.....	73
3.7.8 Prescription Medical Error Detection	74
3.7.9 Pharmacist Recommending Alternative Drugs	75
3.7.10 Vital Signs Healthcare Monitoring	76
3.7.11 Prescription Statistical Analysis.....	76
3.8 IEPO IMPLEMENTATION: IEPO HOSTING AND PRESCRIPTION INFORMATION DISSEMINATION	77
3.8.1 Problems encountered during ontology development and their solutions.	80
CHAPTER 4 FINDINGS AND DISCUSSION: FRAMEWORK EVALUATION	83
4.0 ONTOLOGY CORPUS DOMAIN KNOWLEDGE COVERAGE EVALUATION	84
4.0.1 Key Concept Extraction	84
4.0.2 Enrichment with External Sources	85
4.0.3 Matching and Evaluation.....	85
4.0.4 Corpus-Based Evaluation Metrics	86
4.1 RESULT AND DISCUSSION FOR CORPUS EVALUATION.....	87
4.2 MEDICAL ERROR DETECTION EVALUATION.....	89
4.3 VITAL SIGNS HEALTHCARE MONITORING	90
4.3.1 Body Mass Index: Overweight Alert.....	90
4.3.2 Temperature Monitoring: Covid-19 Alert.....	91
4.3.3 Pulse Rate Monitoring Alerts (Blood pressure follows the same procedure).....	91
4.4 PRESCRIPTION ANALYSIS.....	92
4.5 USABILITY RATING.....	93
4.6 ONTOLOGY EVALUATION USING ONTOLOGY PITFALL SCANNER	95
4.6.1 Classification by Evaluation Criteria.....	96
4.6.2 Classification by Evaluation Criteria: Consistency Metric	97
4.6.3 Classification by Evaluation Criteria: Completeness Metric	101

4.6.4 Classification by Evaluation Criteria: Conciseness Metric	103
4.6.5 IEPO Evaluation using Classification by Dimension.....	103
4.6.6 Classification by Dimension: Functional Dimension Metric	106
4.6.7 Classification by Dimension: Usability-Profiling Dimension Metric.....	107
4.6.8 Summary of OOPS ontology evaluation	110
4.7 IEPO INTEROPERABILITY METRICS AND PROTOTYPE	112
4.7.1 Competency Questions Metric:	112
4.7.2 IEPO Prototype:.....	112
4.7.3 Security: The IEPO prototype	112
4.7.4 Novel Interoperability Index Metric (IA→B):	113
4.7.5 Interoperability Response Time.....	113
4.7.6 IEPO Interoperability Results and Discussion	114
4.7.7 IEPO Prototype and Competency Questions	114
4.7.8 The Interoperability Response Time of IEPO	117
4.8 COMPETENCY QUESTIONS EVALUATION	118
4.9 OTHER IEPO ONTOLOGY EVALUATION METRICS:	119
CHAPTER 5 CONCLUSION AND RECOMMENDATION	121
5.0 INTRODUCTION.....	121
5.1 CONCLUSION OF DISSERTATION BASED ON THE OBJECTIVES.	121
5.2 ANSWERS TO RESEARCH QUESTIONS.....	123
<i>RESEARCH QUESTION 6: What are the most suitable ontology evaluation methodologies, metrics, tools</i> <i>and languages to adopt for evaluating the proposed IEPO framework?</i>	124
5.3 FURTHER RESEARCH RECOMMENDATIONS	126
REFERENCES.....	127

Declaration and Copyright

I, the **Eunice Chinatu Okon**, certify that the dissertation is all my own original work and that I have not obtained a degree in this University or elsewhere on the basis of any of this work.

This dissertation is copyright material protected under the Berne Convention, the Copyright and Neighbouring Rights Act, Act. No. 8 of 2000 and other international and national enactments, in that behalf, on intellectual property. It must not be reproduced by any means, in full or in part, except for short extracts in fair dealing; for researcher private study, critical scholarly review or discourse with an acknowledgement, without the written permission of the office of the Postgraduate School, on behalf of both the author and the BIUST.

Signed:  Date: -----01 FEB 2023-----

Primary Supervisor _____ DR SIGWELE TSHIAMO _____

I, **Dr Sigwele Tshiamo** hereby confirm that I have inspected the above titled thesis, and, to the best of my knowledge, it is based on the original work of the candidate.

Signed:  Date: -----03 FEB 2023-----

Certification

The undersigned certifies that they had read and hereby recommends for acceptance by the Faculty of Science a dissertation titled, AN INTELLIGENT ELECTRONIC PRESCRIPTION ONTOLOGY FRAMEWORK FOR HEALTHCARE INTEROPERABILITY, in fulfillment of the requirements for the degree of master's in Information Systems and Data Management of the BIUST.



Dr. Sigwele Tshiamo

(Primary Supervisor)

Acknowledgements

My deepest and highest gratitude goes to the Almighty God for his mercies, grace, and helps toward the successful completion of this research work.

I am indebted to my supervisors for their patient advice, guidance, and input throughout the research process. Dr. Sigwele Tshiamo and Dr. Galani Malatsi, you have impacted me positively. I also acknowledge and appreciate the effort, support, and encouragement of my Head of Department, Professor Venumadharv Kuthadi just to mention a few.

Family, friends, and colleagues are not left out; your timely moral, financial and other supports remain evergreen. Finally, words will fail to express my profound appreciation to my husband and children, companions in this journey.

Dedication

This work is dedicated to the Almighty God for helping me through this program despite all odds, and to my lovely and beloved husband and children for their unrelenting support and cooperation.

Abstract

Lack of interoperability across healthcare systems has led to high medical costs, delays in healthcare service processes, significant medical errors, and more preventable deaths. Berners-Lee introduced semantic web technology in 2006 to enable communication between different systems and machines with well-defined meanings or semantics using ontologies. The advent of Electronic Healthcare Record (EHR) frameworks such as semantic Electronic Prescription (e-prescription) has been a key turning point in the healthcare domain with the potential to solve healthcare interoperability issues. However, the existing semantic eprescription frameworks have significant limitations, including being proprietary, requiring installation, not web-based, lacking medical error detection and healthcare monitoring, being platform-dependent, incompatibility with other systems, and lacking robust framework evaluation. This research proposes an open-source Intelligent Electronic Prescription Ontology (IEPO) framework for healthcare interoperability to bridge the gaps in existing semantic prescription frameworks. A robust ontology development methodology called Methontology was adopted to develop ontologies for the stand-alone clinic and pharmacy healthcare systems to demonstrate their interoperability. Prescription medical errors like drug interactions are automatically detected during prescribing using Semantic Web Rule Language (SWRL). Healthcare monitoring of vital signs is achieved and demonstrated using SPARQL queries. The proposed IEPO framework was evaluated for any ontology pitfalls regarding consistency, completeness, and conciseness. The results show that none of the ontology pitfalls were detected in the final ontology. IEPO was also evaluated for domain knowledge coverage via the corpus approach and performed 76% better than existing frameworks. IEPO also detected prescription medical errors with higher accuracy than related works. The proposed framework demonstrated seamless healthcare system interoperability between a doctor, pharmacist, and patient by accurately answering all competency questions using SPARQL queries. In addition, the IEPO framework is 12% more usable than related works, with an 87% usability score rating. The IEPO ontology is freely available to download online (shorturl.at/dqGY1).

List of Figures

Figure 2-1. An Illustration of Healthcare Interoperability.....	10
Figure 2-2 Interoperability Level Adopted from [13].	12
Figure 2-3 Comparison of Ontology Development Methodologies.....	18
Figure 2-4 Comparison of Methodology Features	19
Figure 2-5. Percentage of affected ontology languages per issue.....	21
Figure 2-6. Percentage of issues per ontology language	21
Figure 2-7. Percentage of affected ontology development tools per issue (gaps).....	23
Figure 2-8. Percentage of gaps in each ontology development tool	23
Figure 2-9 Issues per tool	25
Figure 2-10 Percentage of tools in literature with.....	25
Figure 2-11 Percentage of gaps in each eprescription framework.....	30
Figure 3-1 Adopted Ontology Development Methodology (Methontology) for the proposed IEPO Framework.....	38
Figure 3-2 IEPO Acquisition, Specification and Conceptualization Activities	41
Figure 3-3 A Concept Classification Tree for the Clinic	49
Figure 3-4 A Concept Classification Tree for the Pharmacy.....	49
Figure 3-5 Clinic Binary Relation Diagram.....	50
Figure 3-6 Pharmacy Binary Relation Diagram	51
Figure 3-7: Merging of clinic and pharmacy concepts classification trees to form the IEPO semi formal ontology.....	65
Figure 3-8. IEPO Formalization Phase	66
Figure 3-9: Creation of Object Properties in Protege	68
Figure 3-10. Creation of Data Properties in Protégé.....	69
Figure 3-11. Linking IEPO Classes Using Object Properties (Relations)	70
Figure 3-12. Creation of IEPO Individuals or Instances by importing the individual axioms from excel to Protégé using Cellfie plugin.	71
Figure 3-13. IEPO Individuals imported to Protégé using Cellfie plugin ad transformation rules.	72

Figure 3-14. Importing and linking IEPO Individuals and relations into Protégé using Cellfie Plugin.....	73
Figure 3-15. 3.8.7 Creating Instances of Data Properties (Attributes) for Imported Individuals: An illustration of importing instances of data properties for the Patient called “Jeda Joel.” ..	74
Figure 3-16. IEPO Implementation Framework	77
Figure 3-17. IEPO GUI.....	80
Figure 4-1 Modified Corpus-based Domain Knowledge Ontology Evaluation Methodology. .	85
Figure 4-2 OOPS Pitfalls level of importance	95
Figure 4-3 The OOPS Landing Page Interface.....	96
Figure 4-4 Evaluation by Categories.....	96
Figure 4-5 Result for consistency metric under classification by evaluation criteria	97
Figure 4-6 Result for pitfalls P06 and P07	97
Figure 4-7 Pitfall P05 result expanded to show the affected ontology elements	98
Figure 4-8 Correction of Pitfall in Protégé: Removing wrong inverse relationships	98
Figure 4-9 Pitfall P19 results expanded to show the effected ontology elements.....	99
Figure 4-10 The protégé Object property	99
Figure 4-11 Pitfall P24 results expanded to show the affected ontology elements.....	100
Figure 4-12 Correction of Pitfall P24 in protégé Removing recursion definition of self-inverse relation and adding symmetric property	100
Figure 4-13 Results for pitfall P04, P12, and P13.	101
Figure 4-14 Pitfall P10 and P11 results expanded to show the affected ontology elements.	101
Figure 4-15 Correction of pitfall P10 in Protégé: Adding Missing Disjoints.....	102
Figure 4-16 Parent Data Properties for Grouping Data Properties for Each Class.	102
Figure 4-17 Results for Conciseness Metric.	103
Figure 4-18 Pitfall Classifications by dimensions	104
Figure 4-19 Results for Modelling Decisions.....	104
Figure 4-20Results for Wrong Inference Metric	105
Figure 4-21 Results for No Inference Metric.....	105
Figure 4-22 Results for Ontology language Metric	105

Figure 4-23 Results for Ontology language Metric	106
Figure 4-24 Results for Real World Modelling Metric.....	106
Figure 4-25 Results for Requirements Completeness Metric	107
Figure 4-26 Results for Application Context Metric.....	107
Figure 4-27 Results for Ontology Clarity Metric	108
Figure 4-28 Correction of pitfall P08 for ontology clarity: Adding annotations	108
Figure 4-29: Results for Real World Modelling Metric	109
Figure 4-30 Results for Ontology Metadata Metric	109
Figure 4-31 Correction of pitfall P08 for Ontology Metadata Metric: Adding license	110
Figure 4-32 The IEPO E-Prescription Prototype	115
Figure 4-33 IEPO SPARQL INSERT Query for recording the Patient's sickness as HIV.	116
Figure 4-34 IEPO prescription retrieval SPARQL Query for Competency Question 1.....	119

List of Tables

Table 2-1 Advantages and disadvantages of various interoperability approaches	13
Table 2-2. A Comparison Between Ontology and Database [2][4]	16
Table 2-3 Ontology Development Methodologies Comparison	17
Table 2-4 Comparison of Ontology Development Languages	20
Table 2-5. A Comparison of Ontology Development Tools.....	22
Table 2-6 Ontology Evaluation tools	24
Table 2-7. A Comparison of eprescription Frameworks	28
Table 3-1 Gaps to be addressed in this research	34
Table 3-2 Glossary of terms of the proposed IEPO Ontology	42
Table 3-3 Class Object Properties Glossary of Terms	43
Table 3-4 Prescription Class Glossary of Terms	44
Table 3-5 Patient Class Data Properties Glossary of Terms	46
Table 3-6 Drug Concept Data Properties Glossary of Terms	46
Table 3-7 Clinic and Pharmacy Properties Glossary of Terms	47
Table 3-8 Disease Concept Data Properties Glossary of Terms	48
Table 3-9 Doctor/Pharmacist/Receptionist Concept Data Properties Glossary of Terms.....	48
Table 3-10 Food/Allergy Concept Data Properties Glossary of Terms	48
Table 3-11 Concept Dictionary	51
Table 3-12 Binary-Relation Table of the IEPO Ontology	54
Table 3-13 Patience Instance Attribute Table.....	55
Table 3-14 Prescription Instance Attribute Table	56
Table 3-15 Drug Instance Attribute Table	57
Table 3-16 Clinic / Pharmacy Instance Attribute Table.....	58
Table 3-17 Disease Instance Attribute Table	58
Table 3-18 Instance Attribute of Doctor/Pharmacy/Receptionist.....	59
Table 3-19 Instance Attribute of Food/Allergy	59
Table 3-20 Class Attribute Table	60
Table 3-21 Formula Table of the IEPO Ontology.....	60

Table 3-22 Logical-Axiom Table	61
Table 3-23 Instance Attribute for the Patient Class	63
Table 3-24 Concept Name	64
Table 4-1b Domain Knowledge Coverage Performance	88
Table 4-2 Proposed IEPO Medical Prescription Error Detection Performance.....	90
Table 4-3 OPPS Evaluation Summary of Classification by Evaluation Criteria	111
Table 4-4 OPPS Evaluation Summary of Classification by Dimension	111
Table 4-5 The interoperability Response Time of IEPO	117

Acronyms

ACRONYM	DESCRIPTION
OWL	Web Ontology Language
RDF	Resource Description Framework
IEPO	Intelligent E-prescription Ontology
SPARQL	SPARQL Protocol and Resource Description Framework (RDF) Query Language
REST	Representational State Transfer (REST) protocol.
DDI	Drug to Drug Interaction
SWRL	Semantic Web Rules Language
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
OCR	Optical Character Recognition
GISPP	Google ImageScraper Python Package
ADE	Adverse Drug Events
API	Application Programming Interface
FDD	Feature Driven Development
RUP	Rational Unified Process
DSDM	Dynamic System Development Method
AM	Agile Modeling
CFM	Crystal Family Methodologies
SDLC	System Development Life Cycle.
OOPS	Ontology Pitfall Scanner
OBO	Open Biomedical Ontology
DINTO	Drug Interaction and evidence Ontology
PDRO	Prescription of Drug Ontology
IRIs	International Resource Identifiers
HER	Electronic Health Record
HTML	Hypertext Markup Language

CHAPTER 1 INTRODUCTION

1.0 Introduction

This chapter provides an overview of the entire research work. This introduction chapter presents the research background, objectives, research questions, research contributions, the significance of the study, relevance to the discipline, evidence of originality, and organization of the thesis.

1.1 Background

Poor or lack of interoperability across healthcare systems has led to high medical costs, delays in healthcare service processes, significant medical errors, and more preventable deaths. In 2006, Bernes-Lee introduced semantic web technology to enable communication between different systems and machines with well-defined meanings or semantics using ontologies. The advent of Electronic Healthcare Record (EHR) frameworks such as semantic Electronic Prescription (e-prescription) has been a critical turning point in the healthcare domain with the potential to solve healthcare interoperability issues. Nevertheless, the existing semantic ePrescription frameworks have significant limitations, including being proprietary, requiring installation, not web-based, lacking medical error detection and healthcare monitoring, being platform-dependent, incompatibility with other systems, and lacking robust framework evaluation. The California Health Care Foundation defined electronic prescription (E-prescribing) as “entering a medication prescription into an automated data entry system such as handheld, PC, or other, and thereby generating a prescription electronically, instead of being written on a piece of paper”. It is a system that enables the transmission of electronically generated prescriptions to pharmacies from the provider’s office. The current Electronic Prescription (e-prescription) frameworks are not interoperable because they are not connected to other healthcare systems like Drugbank [1] which a drug information source. Drug information enables prescription systems to automatically detect prescription errors like

drug interactions, wrong dosage strength, wrong dosage routes, wrong drugs, etc., during the prescribing phase [2][3]. Such medication errors can cause harm to the patient leading to serious side effects, hospitalization, and even fatalities. Prescribers involved can lose their job or license. Annually, 7000 to 9000 Americans die due to medication errors [2] and medication errors are the eighth leading cause of death in the U.S [4]. Globally, the cost associated with medication errors has been estimated at US\$42 billion annually [5]. This research proposes an Intelligent e-prescription Ontology (IEPO) framework for healthcare interoperability between a doctor and a pharmacist. The interoperability of IEPO prevents prescription medical errors like drug to drug interaction, food interaction, disease interactions a wrong prescriptions errors. DDI happens when two or more incompatible drugs are taken together. DDIs are the main reason for death with a 16.6% rate of adverse drug reactions, the primary reason for hospitalization, and are estimated to be the fourth leading cause of death with a significant impact on healthcare costs [6][7]. A wrong medication error is when a doctor prescribes a wrong drug due to confusion between drugs with similar names, inappropriate abbreviations, etc. The study in [8] estimated that 19.3% of medication errors are due to wrong medications. The proposed interoperability will also enable vital signs healthcare monitoring with alerts. The IEPO framework will also enable doctors, pharmacist and patient to share information with each other electronically and remotely. This research proposes an open-source Intelligent Electronic Prescription Ontology (IEPO) framework for healthcare interoperability to bridge the gaps in existing semantic prescription frameworks.

1.2 General Problem

The general problem for the proposed research work is lack interoperability in healthcare systems with a case study of e-prescriptions which cannot be shared between several healthcare systems like the pharmacy, the clinic and the patient electronically . For example, in Botswana, a private clinic cannot share e-prescriptions a pharmacy; neither can a pharmacy revert to the clinic electronically over its e-prescriptions because these systems are standalone and are not interoperable. This lack of interoperability of electronic prescriptions across

healthcare systems has led to high medical costs, delays in healthcare service processes, significant medical errors, and more preventable deaths.

1.3 Problem Statement

A comprehensive literature review was performed to find gaps in current healthcare ePrescription interoperability frameworks. The framework features for current frameworks were examined and it was noted all related works lack more than 80% of vital ePrescription interoperability features where a lack of a feature becomes a gap for that framework. It was noted that with the current frameworks, a patient can't access their prescription data and history electronically. A pharmacist cannot access a patient's prescription data electronically also. Doctors or prescribers cannot access the prescription history of a patient done by other doctors. The pharmacist cannot recommend alternative drugs by requesting the prescriber to alter the prescription for the patient when there is a shortage or due to the cost of a particular prescription drug. It was noted that 82% of frameworks had no Graphical user interface (GUI). None of the frameworks is open source, and all are stand-alone and platform dependent. The current prescription frameworks do not detect prescription medical errors like DDI, food interaction, disease interaction, and wrong prescription drugs. The current frameworks lack interoperability with vital signs healthcare monitoring to give alerts on abnormalities out of standardized range values. None of the current frameworks have been evaluated.

1.4 Research Goal and Objectives

1.4.1 Main Goal

The main goal of this research is to develop an Intelligent E-prescription Ontology (IEPO) framework for healthcare interoperability to overcome limitation with current electronic healthcare interoperability frameworks.

1.4.2 Objectives

1. To conduct a comprehensive literature review on semantic e-prescription interoperability frameworks to determine the existing gaps.

2. To conduct a comprehensive literature review to find the most suitable ontology development methodologies, tools, languages, and suitable ontology evaluation methodologies, tools, and metrics to adopt for IEPO development.
3. To create an electronic prescription ontology called IEPO for healthcare interoperability.
4. To create a SPARQL endpoint Application Programming Interface (API) to enable various healthcare stakeholders like doctors, pharmacists, and patients to access IEPO healthcare data seamlessly and remotely via the internet.
5. To evaluate the performance of the proposed IEPO framework.

1.5 Research Questions

1. What are the gaps found in current healthcare interoperability frameworks from literature based on semantic e-prescription to be addressed by the proposed IEPO framework?
2. What are the most suitable healthcare interoperability types and approaches to adopt for developing the IEPO framework?
3. What are the most suitable interoperability standards to adopt for the development of the IEPO framework?
4. Between ontologies and databases, which is the most suitable to adopt for developing the IEPO framework?
5. What are the most suitable ontology development methodologies, languages, and tools to adopt for the development of the IEPO framework?
6. What are the most suitable ontology evaluation methodologies, metrics, tools, and languages to adopt for the evaluation of the proposed IEPO framework?
7. How can ontologies enable healthcare interoperability in the proposed framework?
8. Will the proposed ontology be open source and be made available online for free download?
9. How will doctors, patients, and pharmacist share information with each other in the proposed framework?

10. How well does the proposed IEPO framework perform compared to other related works?

1.6 Research Contribution

1.6.1 Research Papers

- Article Submitted to International Journal of Semantic Computing; Article Title: Intelligent Semantic e-prescription for Healthcare Systems Interoperability to Detect Medical Error with Corpus-Based Domain Knowledge Ontology Evaluation.
- Article Submitted to Journal of Web Semantics: Article Title: Development and Evaluation of an Intelligent e-prescription Ontology for Healthcare Interoperability and Medical Error Prevention.
- Conference paper Accepted and submitted to (Botswana international University of Science and Technology) TRIADS 2022 Article Title: Interoperability of Botswana Public and Private Healthcare System.

1.6.2 Research Work Contributions

1. Developing an open source e-Prescription interoperability framework using ontologies called IEPO to seamlessly share e-Prescription information between several healthcare entities like doctors, pharmacists, patients, medical aid, research, machines, etc.
2. The IEPO ontology is open source and was made freely available online to download for reuse to further improve research in the domain. Download link: shorturl.at/dqGY1
3. Prevention of e-Prescription medical errors during prescribing phase by detecting prescription errors like drug interactions and wrong prescription errors during prescribing using Semantic Web Rule Language (SWRL).
4. Real-time pharmacy recommendations to the prescriber to alter prescription and opt for alternative drugs for patients due to drug unavailability or cost.

5. Evaluation of the IEPO framework in terms of domain knowledge coverage performance, medical error detection accuracy, competency questions, interoperability index, and Interoperability Response time.
6. Proposing a novel and quantifiable semantic interoperability evaluation metric called "Interoperability Index" between two systems.

1.7 Significance of study

This research work is significant because human health and safety is crucial. The presented framework significantly reduces harmful effects caused by prescription medical errors, which can save lives. The IEPO framework has a prototype and an IEPO ontology as a product. The open source ontology made available for free download online (Shorturl.at/dqGY1) can be used for further research in areas of semantic interoperability, ontology development, ontology evaluation, and medical error preventions. The IEPO framework can be deployed by healthcare entities like government, hospitals, insurance companies and even machines since machines understand OWL languages. Further significances of study can be extracted from the entire thesis.

1.8 Relevance to the Discipline

This work finds relevance to the research disciplines as follows:

Ontologies and Semantic Web: This discipline is very vast and the research's relevance is in ontology development and evaluation. The developed and freely available IEPO ontology is very relevant to the semantic web domain to improve research in the area as it can be downloaded, reused and integrated with other ontologies to further improve the health of humanity. The proposed semantic web rule languages for medical errors prevention can be improved by researchers and incorporated in future frameworks to save lives.

Healthcare Interoperability: The research in this discipline is to save lives, cut down on healthcare cost and improved on time constraints of healthcare processes. The proposed prototype with a GUI is relevant to the healthcare interoperability domain and its opensource, which mean reseahers can reuse it and improve on it to significantly solve the healthcare interoperabilit issues. The proposed prototype is compatible with any healthcare system such that any research with a preliminary system or prototype can intergrate the IEPO framework with thir own seamlessly.

1.9 Evidence of originality

The signed “Declaration and Copyright” form at the beginning of this thesis justifies that this work by done by Eunice Okon. The adopeted methods, tools and ideas were cited in this reseahc work.

1.10 Organization of the Dissertation

The rest of this dissertation is organized as follows.

Chapter 2: This chapter presents the background on healthcare interoperability, types of healthcare interoperability, the benefit of healthcare interoperability, the challenges of healthcare interoperability, interoperability standards, literature review on ontology development languages, tools, methodologies. Also, semantic electronic prescriptions frameworks are introduces the gaps with current healthcare interoperability frameworks are presented and analysed.

Chapter 3: This chapter presents the proposed framework development based on ontology development, the adopted methodology framework (Methontology), and proposing an intelligent semantic e-prescription framework called IEPO which prevents medical errors in healthcare systems.

Chapter 4: In this chapter, the performance of the proposed e-prescription ontology framework was evaluated using the corpus domain knowledge to ascertain the coverage of the e-prescription domain knowledge. Other evaluation methods used included Ontology Pitfall Scanner to test for quality and correctness, and competency questions evaluation. Other evaluation methods are also employed with their metrics.

Chapter 5: This chapter summarizes the research work, highlights findings, and then presents recommendations for further research directions.

CHAPTER 2 LITERATURE REVIEW

2.0 Introduction

This chapter aims to answer the research questions presented in Section 1.7 by conducting a comprehensive literature with a summary justification of the best options (methodologies, tools, languages, frameworks, evaluation metrics, etc.) to adopt for IEPO development from the literature. The chapter will define healthcare interoperability and present its different types (Section 2.3) and approaches (Section 2.4) followed by a justification summary of the best approach to adopt for IEPO development. Various healthcare interoperability standards will be presented in Section 2.5 followed by a justification summary of the most suitable standard to be adopted in Section 2.5. This chapter will also compare ontologies and databases in Section 2.6.1 and give a justification for choosing ontologies. In addition, related works and a summary analysis of ontology development methodologies, languages, and tools will be presented in Section 2.6 followed by the justification of the adopted choices for IEPO development. The chapter will also present current interoperability frameworks from the literature in Section 2.7 based on semantic e-prescription and their gaps. Finally, in Section 2.7.2, the chapter will present a literature review and summary analysis of ontology evaluation methodologies, metrics, and tools with the justification of the most suitable to adopt for IEPO framework evaluation.

2.1 Healthcare Interoperability

According to [7] – [9], healthcare interoperability is the ability of two or more sub-systems of the healthcare information system to communicate, share, interpret and use data reliably to promote accessibility and avoid contingency in the healthcare system. Interoperability is a fundamental requirement for the improvement of healthcare delivery [10] and it is illustrated in Figure 2-1. The following benefits can be realized through healthcare interoperability [11-19]

improved patient safety and satisfaction, reduced medical errors, reduced cost due to reduced medical errors and reduced use of paperwork, easier accessibility to patient data, better analysis of chronic disease, a better understanding of medical terms, and health systems collaboration due to integration of records. The following benefits can be realized through healthcare interoperability -[9] - improved patient safety and satisfaction, reduced medical errors, reduced cost due to reduced medical errors and reduced use of paperwork, easier accessibility to patient data, better analysis of chronic disease, a better understanding of medical terms, and health systems collaboration due to integration of records.

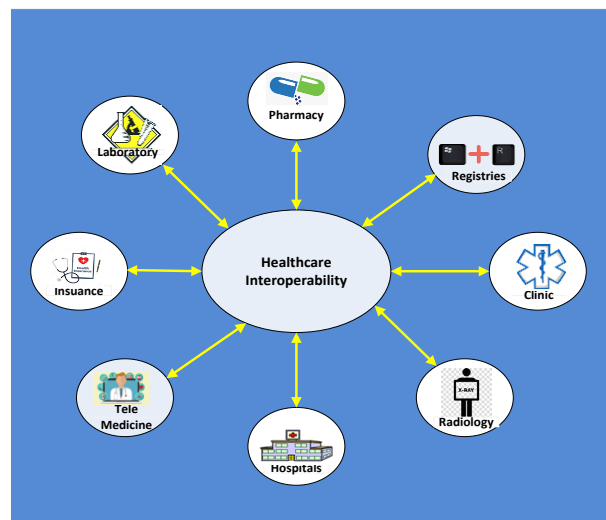


Figure 2-1. An Illustration of Healthcare Interoperability.

The following are the challenges for healthcare interoperability that need to be addressed [9][10] - complexity due to heterogeneous actors involved, unstructured big data, different healthcare standards interpretations, incompatible terminologies and vocabularies used by healthcare systems with different concept representations, poor adoption of Electronic Health Records (EHR) where paperwork is still being used, etc. According to [11] are many levels of healthcare interoperability technical interoperability which requires communication channels and protocols for data transmission, syntactic interoperability which is how data is communicated between components, semantic interoperability which ensures that the

meaning of medical concepts can be shared and interpreted across systems and organizational interoperability involves organizations, legislation, and policies. This research will focus on semantic healthcare interoperability using EHR, specifically e-prescription. Semantic interoperability is advantageous over others because according to [9], it is the future of healthcare. In addition, the author in [12] postulated that semantic interoperability is advantageous from the rest because it ensures that the receiving system understands the meaning of the exchanged information, even when the algorithms used by the receiving system are unknown to the sending system.

2.2 Types/Levels of Healthcare Interoperability

The exchange of information between healthcare systems can happen at varying levels of interoperability according to [13] These levels are as follows:

2.2.1 Foundational interoperability:

This level of interoperability deals with information exchange between standalone systems. It defines the requirements for one system or application to securely inter-connect with another system, to send and receive data. It requires the end user or other technology intervention for the receiving system to interpret the received data.

2.2.2 Structural Interoperability:

This level of interoperability also referred to as syntactic interoperability, defines the syntax of data exchange between two or more systems using a common interoperability protocol such as the High-Level Architecture (HLA) to provide meaningful information to the end users and it also involves data modeling and Data structure. It ensures that data exchanges between information technology systems can be interpreted at the data field level [14]

2.2.3 Semantic Interoperability:

This level of interoperability enables multiple systems to exchange, interpret and use shared information in a similar way. It takes advantage of both the structuring of the data exchange and the codification of the data, including standard, publicly available vocabulary, so that the receiving information management systems can interpret the data. Semantic interoperability supports the electronic exchange of patient data and information among authorized parties via

potentially disparate health information and technology systems and products to improve the quality, costs, safety, efficiency, experience, and efficacy of healthcare delivery.

2.2.4 Organizational Interoperability

Organizational interoperability enables seamless and secure integration of business processes and workflows within and across the boundaries of organizations. To complement the other three interoperability levels, the wiliness and commitment of concerned individuals and organizations to cooperate in sharing information at this level determine the efficiency and overall success of the whole process.

This research work will focus on semantic interoperability because it has a concept that makes information that is exchanged meaningfully to the end users; it also contains web ontology and other Resources Description Framework (RDF) and SPARQL protocol and RDF Query protocol.

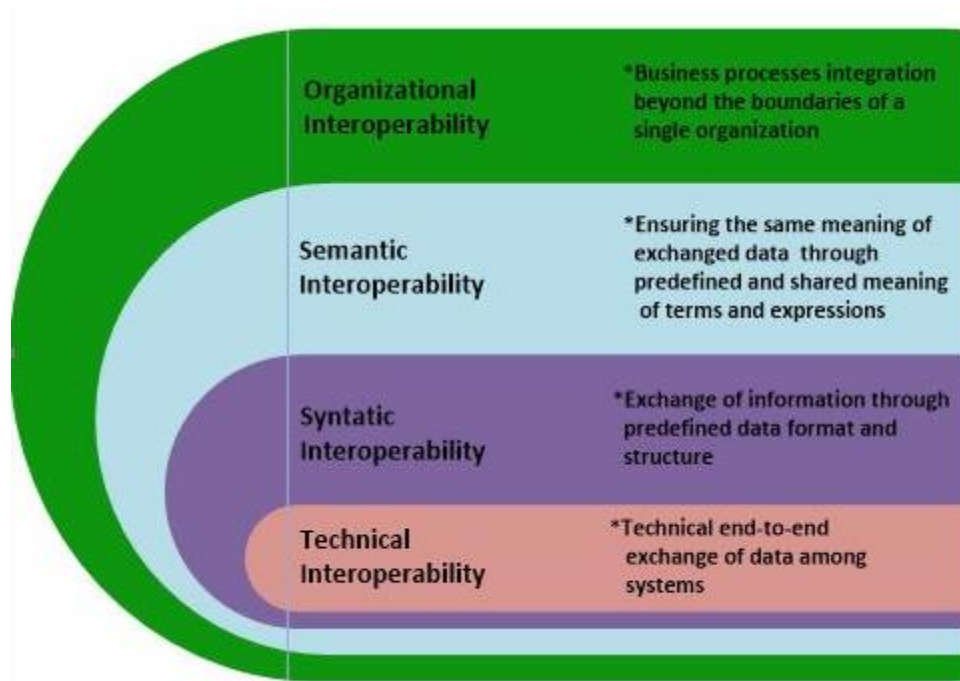


Figure 2-2 Interoperability Level Adopted from [13].

2.3 Interoperability Approaches:

For securing data and managing semantic interoperability, some approaches can be used which are as follows [15],[16]:

- Layered Approach
- Centralized Approach
- Decentralized Approach
- Similarity-based Approach.

Table 2-1 shows the advantages and disadvantages of various interoperability approaches. This research work will adopt the centralized approach with justification that data is managed and controlled easily from generic model independent of any specific architecture, also, centralized models are highly effective in terms of sharing data because data is made available in one place and can be accessed by several entities easily, the drawback might be single point of failure but with advance failover techniques where backups are made, a centralized approach's single point of failure is eliminated.

Table 2-1 Advantages and disadvantages of various interoperability approaches

Approaches	Advantages	Disadvantages
Layered Approach: Used to reduce the gap between syntactic and semantic interoperability levels	It is based on the classification of inputs into syntax, object, and semantic attributes. Decomposition of tasks gives ease in functionality and reduces the complexity of execution. The functionality of each layer is independent of the other.	A mapping is required between layers to generate a consolidated output or decision on the given input size. It is a viable approach in achieving interoperability but needs to address integrity issues while sharing EHR across multi-platform environments
Centralized Approach: Data is managed and controlled from a single place or location by creation of generic model independent of any specific architecture	It manages and controls sharing of EHR in flexible client-server environment. Interoperability can be achieved well if the collaborating units are kept to minimum	With the increase in the number of units, interoperability becomes highly complex and due to the flexible nature of the approach, sensitive health data becomes highly vulnerable to security breaches.
Decentralized Approach:	It exhibits high security because no direct access is	Though, highly robust security parameter, decentralized

In this approach, each system maintains its own source of data model and architecture independently.	permitted to each other databases.	approach falls low on interoperability as no EHR-system mutually integrates its functionality with other.
Similarity-Based Approach: Confidence measure is established between two elements in different user hierarchies. The similarity is expressed in a mathematical number that typically ranges from [0-1].	This approach maintains a balance between interoperability and security of EHR while sharing between independent healthcare users.	Can be complex to implement.

2.4 Interoperability Standards

To achieve interoperability in healthcare, several data standards and rules have been developed to enable exchange usable data between different and isolated IT systems and applications. Some of the standards used in healthcare interoperability are:

Health Level Seven (HL7) – It was developed by Health Level Seven international as a set of international standards that provides a framework for sharing and exchange of clinical and administrative data between various healthcare providers’ software applications. They are the most widely used sets of messaging standards and function in the OSI model application layer (layer 7) [17] HL7 suite include the following standards according to [18]

HL7 version 2 (v2) – It is the most extensively used messaging standard which provides messaging specifications for sharing of clinical information such as laboratory test request and results, patients’ appointments, transfer, referrals and discharge, and pharmaceutical prescriptions. **HL7 version 3(v3)** – It was developed to address some of the limitation and improve upon the HL7 version 2. It functions with increased detail, clarity and precision compared as compared to version 2. **Reference Information Model (RIM)** and **HL7 Development Framework** are the information model and formal methodology used by HL7 V3. **Clinical Document Architecture (CDA)** – This standard is based on HL7 version 3, it defines the structure of certain medical documents which can be in the form of text, images, etc. These

documents include discharge summaries, progress notes and referral forms. FHIR (Fast Healthcare Interoperability Resources) – The latest standard proposed by HL7. It is a standard for the exchange of resources (modular components) [19].

SNOMED-CT (Systemised Nomenclature of medicine for Clinical Terminologies) – SNOMED-CT was developed by SNOMED International. It was formed by the merging of two healthcare terminologies - UK Clinical Terms version 3 (also called Read Codes) with SNOMED RT (Reference Terminology). It enables clinical free text to be represented in a semantically rich, machine-readable manner [20].

Logical Observations, Identifiers, Names and Codes (LOINC) -This is a standard for identifying clinical information (tests and observations) in electronic laboratory reports. It was developed by Regenstrief Institute, Indiana as Laboratory Observations, Identifiers, Names and Codes (LOINC). The term laboratory was replaced by logical to include non-laboratory observations such as vital signs or electrocardiograms. LOINC improves the reliability of HER, improves information exchange and integration of healthcare networks. It gives a widespread code framework that incorporates a set of names and ID codes for distinguishing clinical laboratory information and test results [21].

DICOM (Digital imaging and communication in medicine)

This standard was first proposed by the National Electrical Manufacturers Association (NEMA) and the American College of radiology [survey of DICOM]. It is the leading standard for capturing, storing, exchange, printing and handling of image data in medical applications. In a PACS (Picture Archiving and Communication System), integration of printers, scanners, servers and other network devices from different vendors is enabled by DICOM [overview of HC Inter, 2013]

Open EHR (Electronic health records standards for clinical information and patient records)

The Open HER Foundation developed this standard, it is a standard for enabling semantic interoperability of electronic health records and clinical information systems. [overview of HC Inter, 2013]

2.5 Ontology

Ontologies describe a set of concepts and their relationship in a specific domain. They are necessary for knowledge representation and knowledge exchange, in digital libraries, semantics web, and personal information management. More simply, an ontology is a way of showing the properties of a subject area and how they are related, by defining a set of concepts and categories that represent the subject [22], [23].

2.5.1 Comparison Between Ontologies and Databases

Table 2-2 shows a comparison between ontologies and databases. This research deals with the use of ontologies for healthcare interoperability and comparison of ontologies with databases which justify why ontologies are adopted than databases because of their strengths

Table 2-2. A Comparison Between Ontology and Database [2][4]

Item	Ontology	Database
Focus	-Preservation of meaning to facilitate interoperability -Focus on global semantics of the real world	-Database schema stores and queries large data sets -Focus on local semantics that has only aspects of the real world
Design Approach	Ontologies are dependent on semantics so can be created by reusing existing ontologies	Created from scratch for a specific purpose and are not generally reusable
Manner of knowledge representation	Uses open world assumption (OWA)	Relies on closed world assumption (CWA), which assumes that the model represents the complete information
Syntax	Uses logic to describe the syntax. OWL, RDF languages, and Semantic features are mandatory	Uses entity-relationship diagrams to describe the syntax, Semantic features unimportant.
ISA hierarchy	ISA hierarchy is a backbone	Little emphasis on the ISA hierarchy

Optimization	Ontology-creating methodologies consist of patterns	Normal forms are used for database system
--------------	---	---

2.5.2 Ontology Development Methodologies

Several ontology development methodologies will be reviewed to choose the most suitable one for IEPO development. Ontology development methodologies refer to a set of guiding principles on how the identified activities in the process of ontology design are duly carried out [24]. [24] According to literature, [25] some of the popular ontology development methodologies used by developers include Ushcold and King, Gruninger & Fox, Methontology, On-To-Knowledge, Senus, Kactus, Diligent, Hcome, Noy McGuinness & Method, UPON, and Agile development. Table 2-33 compares these methodologies in terms of several features. A lack of a feature is denoted as a gap in that framework.

Table 2-3 Ontology Development Methodologies Comparison

Features	Ushcold and king [16]	Gruninger & Fox (TOVE) [17]	Methontology [18]	On-To-Knowledge [19]	Senus [19]	Kactus [19]	DILIGENT [20]	HCOME [21]	NOY MCGUINNESS Method [22]	UPON [23]	% of methodologies per feature issue
Sufficient Details	0	0	✓	0	0	0	✓	✓	0	✓	60%
Life-cycle recommendation	0	0	✓	✓	0	0	✓	✓	0	✓	50%
Pre-development process proposed	0	0	0	✓	0	0	✓	0	0	0	80%
Required process explained	0	✓	✓	✓	0	0	✓	0	0	✓	50%
Designed process explained	0	✓	✓	✓	0	✓	✓	0	0	✓	40%
Implementation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0%

<i>Past Development Activities</i>	0	0	✓	✓	0	0	✓	✓	0	0	60%
<i>Documentation described</i>	0	0	✓	✓	0	0	✓	✓	0	0	60%
<i>Configuration Management details</i>	0	0	✓	0	0	0	0	0	0	0	90%
<i>Knowledge Acquisition details</i>	0	0	✓	0	0	0	0	✓	0	✓	70%
<i>Evaluation Details</i>	0	0	✓	✓	0	0	0	0	0	✓	70%
<i>Distributed Working</i>	0	0	0	0	0	0	✓	✓	0	0	80%
<i>Project Management Activities</i>	0	0	✓	✓	0	0	0	0	0	0	80%
<i>Supporting Tools</i>	0	0	✓	✓	✓	0	✓	✓	✓	✓	30%
% of feature issues per Methodology	93%	79%	14%	29%	86%	86%	29%	43%	86%	43%	

2.5.3 Summary of Ontology Development Methodology

The proposed research will adopt Methontology because it has many useful features compared to methodologies, lacking only 14% of the useful features required in ontology development tools. Methontology is also used because a lot of research uses it, it is easy to follow, and several publications have adopted Methontology. Figure 2-5 and Figure 2-6 analyses Table 2-3 further to validate the justification for adopting methontology.

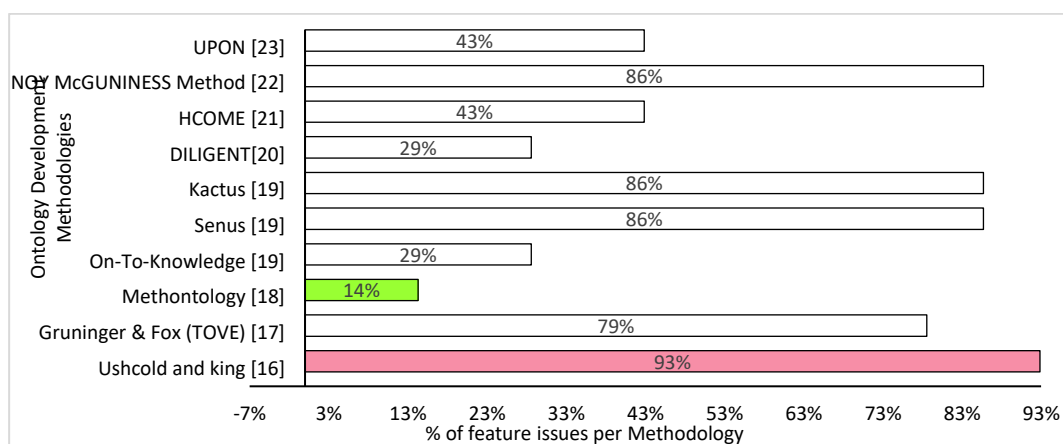


Figure 2-3 Comparison of Ontology Development Methodologies

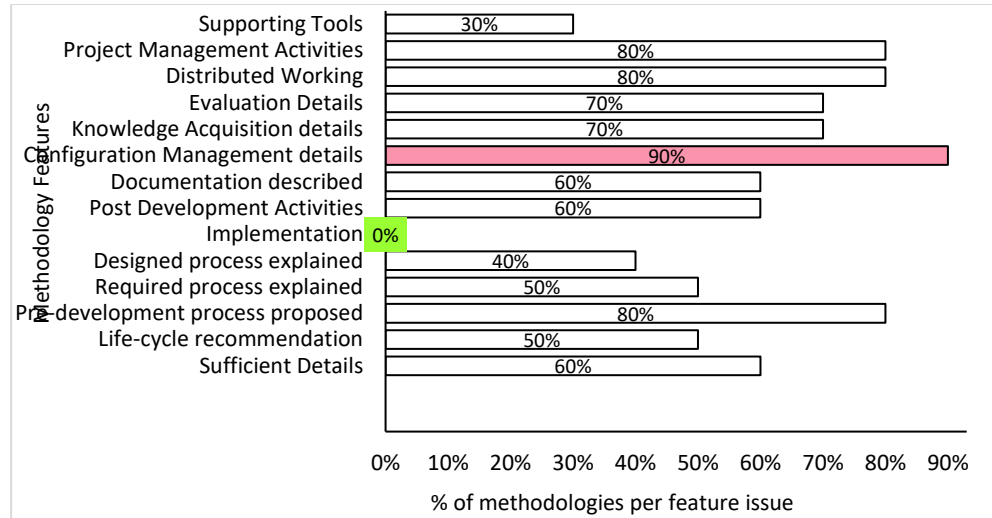


Figure 2-4 Comparison of Methodology Features

2.5.4 Ontology Languages

Several ontology languages will be reviewed to choose the most suitable one for IEPO development. Table 2-5 shows a comparison of ontology development languages.

Ontology languages are formal languages used to construct ontologies, several ontology languages have been developed, and they become ontology languages adopted in the context of the Semantic Web [32]. Table 2.2 shows the ontologies language comparison which includes Loom, OCML, FLogic, SHOE, OML, XOL, RDF(S) OIL, DAML+OIL, OWL, and OWL2 and their general issues, Concept partitions, documentation attribute, Instance attributes, Class attributes, Local scope Facet, Default slot value, Type constraints, Cardinality constraints, Slot documentation Taxonomies, the subclass of, Disjoint decomposition relations, n-array relations, Operational definition Axioms, 1st order logic, 2nd order logic, Named axioms, Embedded axioms instances, Concepts instances, Facts other and rules.

Table 2-4 Comparison of Ontology Development Languages

Ontology Issues	Ontology Development Languages											% of Languages per issue
	LOOM	OCML	Flogic	SHOE	OML	XOL	RDF(s)	OIL	DAML + OIL	QWL	OWL2	
Concept partitions	✓	X	X	X	✓	X	X	✓	✓	X	✓	55%
Documentation	✓	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	9%
Instance attributes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0%
Class attributes	✓	✓	✓	X	✓	✓	X	✓	✓	✓	✓	18%
Local-Global scope	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0%
Global scope Facts	✓	x	x	X	✓	✓	✓	✓	✓	✓	✓	27%
Default slot value	✓	✓	✓	X	X	✓	✓	✓	✓	✓	✓	18%
Type constraints	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0%
Cardinality constraints	✓	✓	X	X	X	X	X	✓	✓	✓	✓	45%
Slot documentation	✓	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	9%
Subclass of	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0%
Disjoint Decomposition	✓	✓	X	X	X	X	X	✓	✓	X	✓	55%
n-ary relations	✓	✓	X	✓	✓	X	X	X	X	X	✓	55%
Operational definition	✓	✓	✓	X	X	X	X	X	X	✓	✓	55%
1st order logic	✓	✓	✓	X	✓	X	X	X	X	✓	✓	45%
2nd order logic	X	X	X	X	X	X	X	X	X	✓	✓	82%
Named axioms	X	✓	X	X	X	X	X	X	X	X	✓	100%
Embedded axioms	✓	✓	X	X	✓	X	X	X	X	✓	✓	55%
Concept instances	✓	✓	✓	✓	✓	X	✓	✓	✓	✓	✓	0%
Facts	X	X	X	X	✓	X	X	X	X	✓	✓	73%
Rules	✓	✓	X	✓	✓	X	X	X	X	✓	✓	45%
% of issues per language	14%	19%	57%	57%	29%	57%	57%	38%	38%	19%	0%	

2.5.5 Summary of Ontology Language

The proposed research will adopt the OWL2 language for ontology development with justification that OWL2 has 0% of ontology issues compared to others. Figure 2-5 and Figure 2-6 analyses Table 2-5 further to validate the justification for adopting OWL2. OWL2 supports several features compared to others like concept instances, facts, rules, based on instances, 1ST order logic, 2nd-order logic embedded axioms, Operational definition, subclass of, disjoint decomposition, concept partition and also it allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary and complete reasoning.

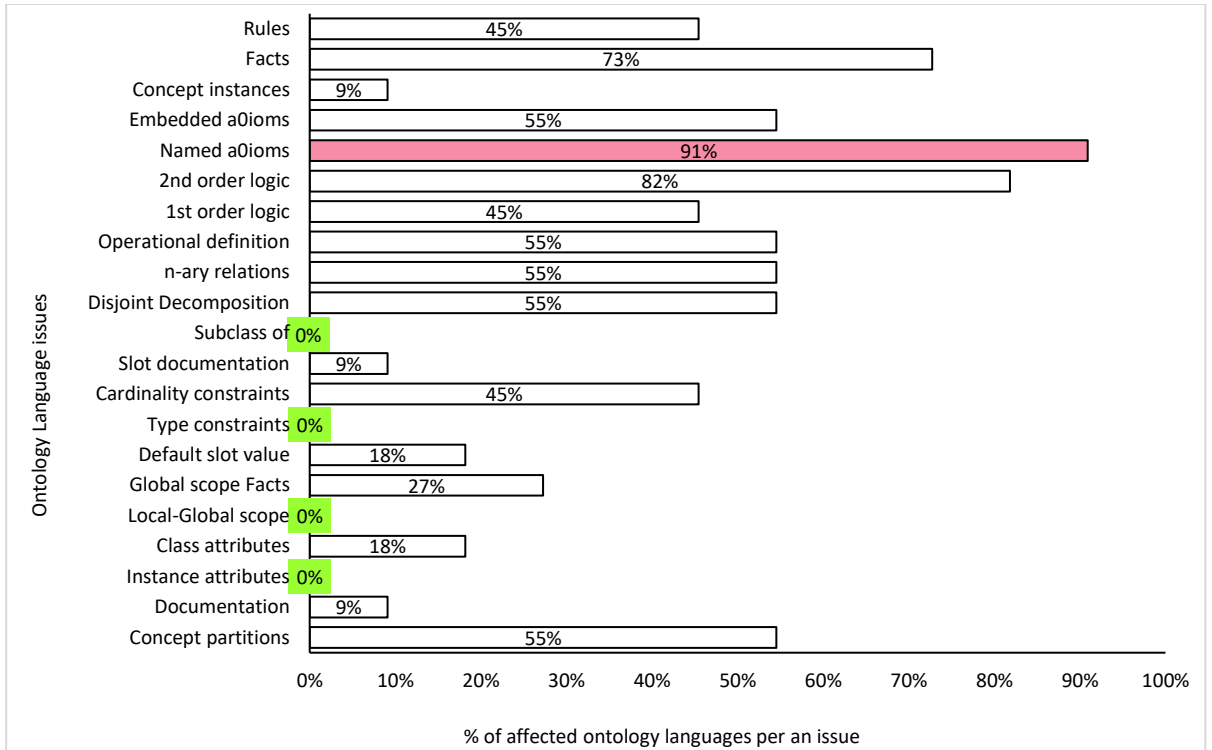


Figure 2-5. Percentage of affected ontology languages per issue

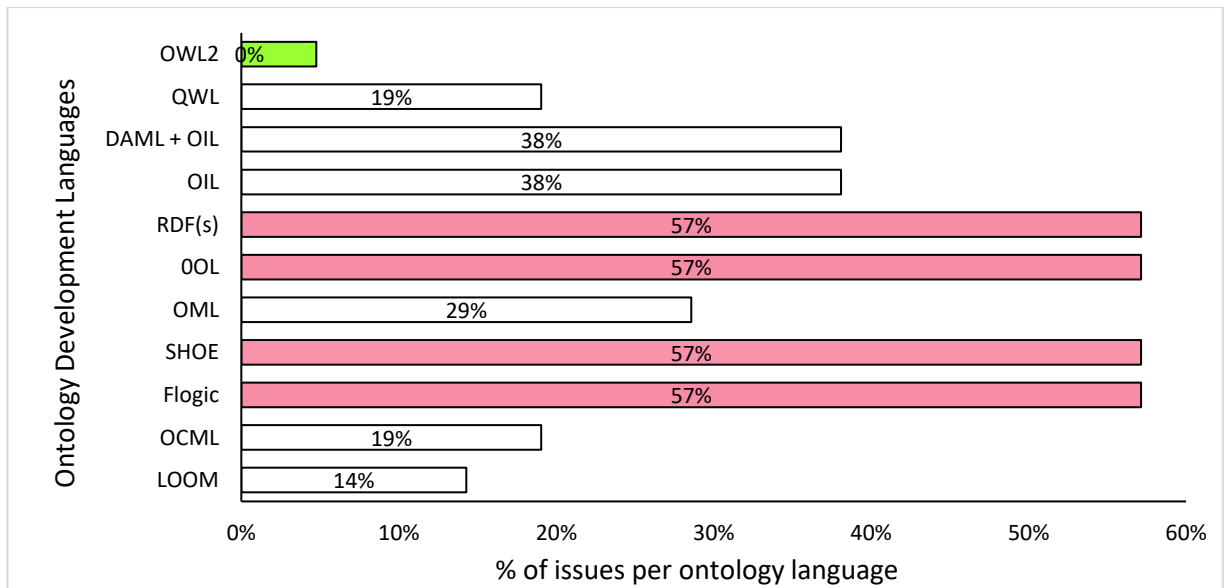


Figure 2-6. Percentage of issues per ontology language

2.5.6 Ontology Development Tools

Several ontology development tools will be reviewed to choose the most suitable one for IEPO development. Table 2-6 shows the different ontology development tools and their features,

The tools include the protégé, OntoEDIT(Free), DOE, IsaViz, Ontolingua, Altova Semantic Works, OilEd, WebODE, Powl, SWOOP, TopBriadComposer, Neon Toolkit, Morta, OBO-Edit, Hozo, OntoBuilder/Ontoedit, WSMO Studio, TODE, OWLGrED, and their features which includes Free, opensource, Java-based, extensibility, Collaboration, Architecture, import languages, exports languages, and tools.

Table 2-5. A Comparison of Ontology Development Tools

Ontology Development Tools	Vital Ontology Development Tool Features								% of issues per tool
	Free	Open source	Java-based	extensibility	Collaboration	Community Support	OWL2 support	Reasoner	
<i>Protege [26]</i>	✓	✓	✓	✓	×	✓	✓	✓	13%
<i>OntoEdit (Free)</i>	✓	×	✓	✓	×	×	×	×	63%
<i>DOE</i>	✓	×	✓	×	×	×	×	×	75%
<i>IsaViz [27]</i>	✓	✓	✓	✓	×	×	×	✓	38%
<i>Ontolingua</i>	✓	×	×	×	✓	×	×	✓	63%
<i>Altova Semantic</i>	×	×	×	×	×	×	×	✓	88%
<i>OilEd</i>	✓	✓	✓	×	×	×	×	✓	50%
<i>WebODE [24]</i>	✓	×	✓	✓	✓	×	×	✓	38%
<i>Powl</i>	✓	✓	×	✓	✓	×	×	✓	38%
<i>SWOOP [28]</i>	✓	✓	✓	✓	✓	×	×	✓	25%
<i>TopBriad Compos [29]</i>	×	×	✓	✓	✓	×	×	✓	50%
<i>Neon Toolkit</i>	✓	✓	✓	✓	✓	×	×	✓	25%
<i>Morta</i>	✓	✓	×	✓	×	×	×	✓	50%
<i>OBO-Edit</i>	✓	✓	✓	✓	×	×	×	✓	38%
<i>Hozo [30]</i>	✓	✓	✓	×	✓	×	×	✓	38%
<i>OntoBuilder</i>	✓	✓	✓	✓	×	×	×	✓	38%
<i>WSMO Studio</i>	✓	✓	✓	✓	×	×	×	✓	38%
<i>TODE [30]</i>	✓	×	×	×	×	×	×	✓	75%
<i>OWLGrED[31]</i>	✓	✓	✓	✓	✓	×	✓	✓	13%
% of tools per issue	11%	37%	26%	32%	58	95%	89%	11%	

2.5.7 Summary of Ontology Development Tools

The proposed research will adopt the Protégé tool for ontology development with justification that Protégé has 13% of issues compared to others. Figure 2-7 and Figure 2-8 analyses Table 2-6 further to validate the justification for adopting Protégé. Protégé is mainly open source and is used by many researchers with a lot of community support.

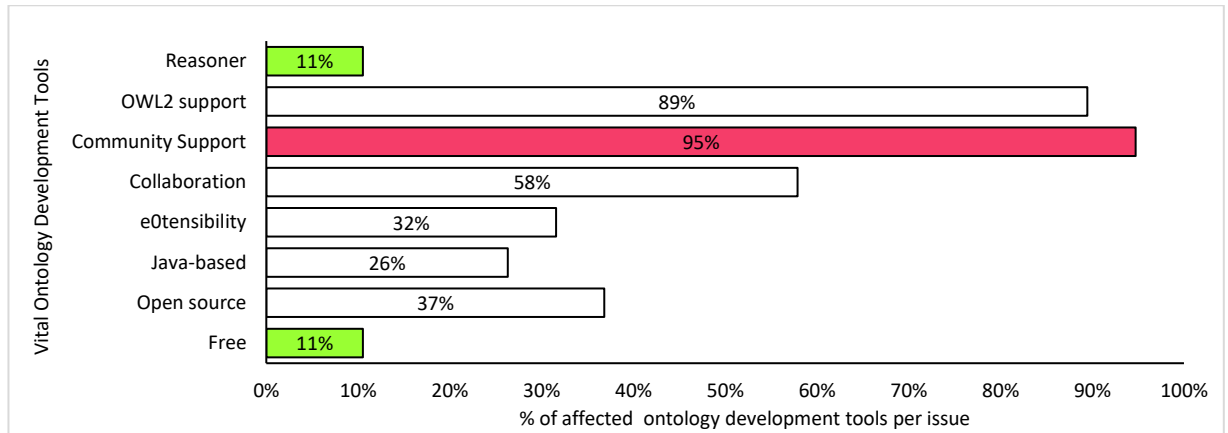


Figure 2-7. Percentage of affected ontology development tools per issue (gaps)

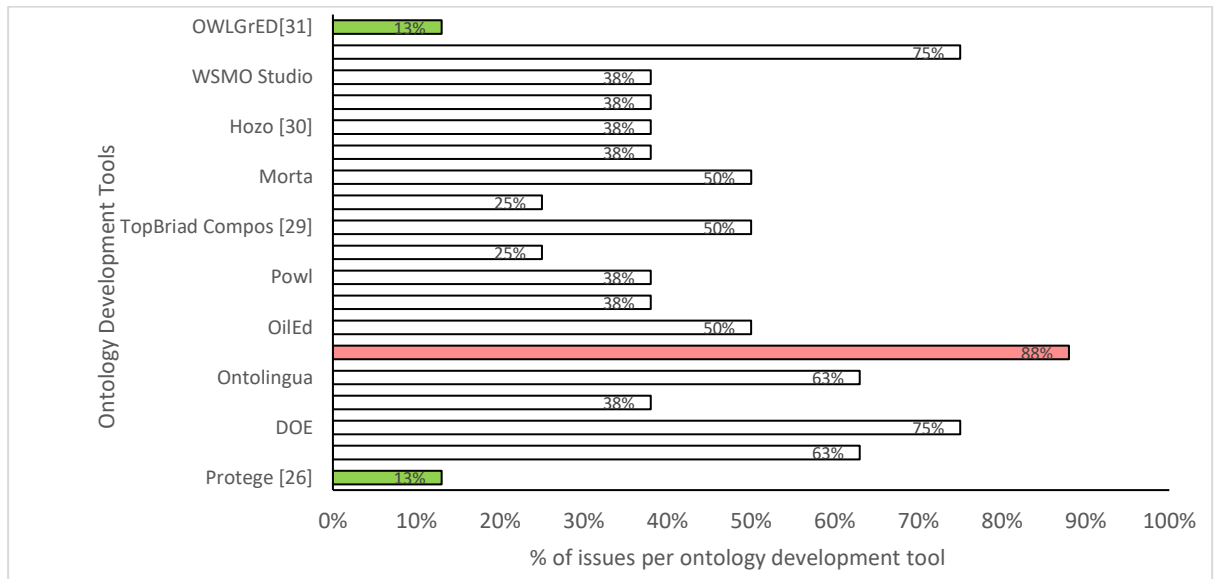


Figure 2-8. Percentage of gaps in each ontology development tool

2.5.8 Ontology Evaluation Tools

Table 2-7 shows a comparison of commonly used tools in research for evaluating ontologies and their usability characteristics. These tools can be categorized as Web-based, which are

OOPS! Moki, OQuaRE, Plug-ins like XD- Analyzer, Onto Check; and finally, desktop or API based like Eyeball. The tick represents the presence of the usability characteristic of the tool presented.

Table 2-6 Ontology Evaluation tools

Features	Web-based			Plug-in		Desktop API	% of tools per feature issue
	<i>OOPS</i>	<i>Moki</i>	<i>OQuaRE</i>	<i>XD - Analyzer</i>	<i>Onto - Check</i>	<i>Eyeball</i>	
<i>GUI</i>	✓	✓	✓	✓	✓	x	16%
<i>API</i>	✓	x	✓	x	x	✓	50%
<i>No installation process</i>	✓	x	✓	x	x	x	60%
<i>ontology editor independence</i>	✓	x	✓	x	x	✓	50%
<i>Custom evaluation</i>	✓	x	X	x	x	✓	60%
<i>Offline use</i>	x	✓	X	✓	✓	✓	30%
% of issues per tool	16%	60%	30%	60%	60%	30%	

Summary of Ontology evaluation tools: The proposed research will adopt the OOPS tool for ontology evaluation with justification that OOPS has 16% of issues compared to others. Figure 2-7 and Figure 2-8 analyses Table 2-6 further to validate the justification for adopting Protégé. The justification of the choice of the proposed ontology evaluation tool (OOPS!) is that it meets many usability characteristics compared to others as it is open source, requires no installation, has a user-friendly web-based GUI that is customizable, and also used in detecting ontology pitfall errors and also give solution on how the pitfall errors will be corrected.

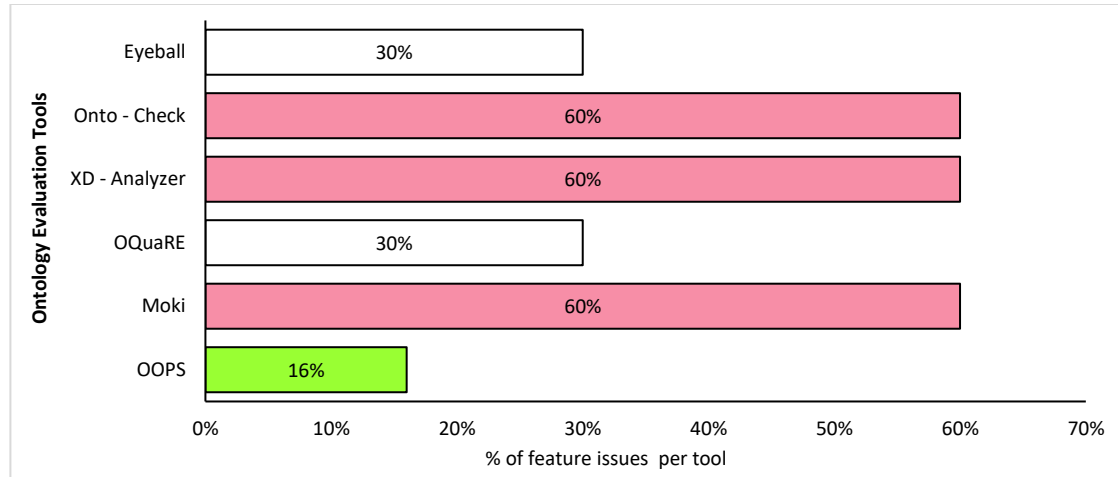


Figure 2-9 Issues per tool

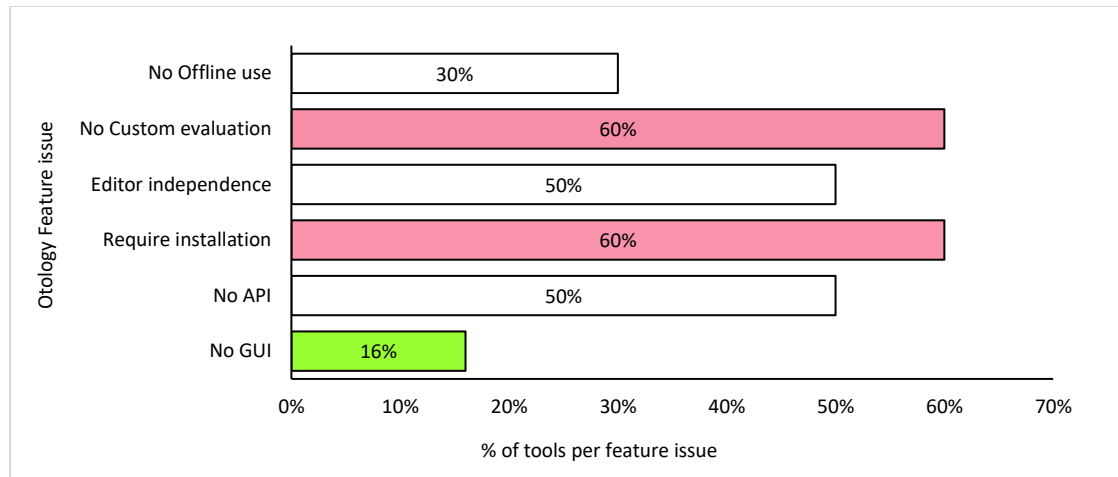


Figure 2-10 Percentage of tools in literature with

2.6 Healthcare Interoperability Using Semantic e-Prescription

2.6.1 E-Prescription Review

An e-prescription system is an EHR healthcare system that facilitates the interaction between physicians and pharmacies by enabling physicians to create and pass on prescriptions electronically to pharmacies [[39]. E-prescriptions are being implemented in clinical settings to

reduce medication-related risks and enhance patient safety [40]. A semantic e-prescription uses ontologies to facilitate domain knowledge sharing, with common vocabulary across independent healthcare entities like the clinic, pharmacy, drug sources, patient, etc. The main advantages of semantic e-prescription systems are as follows:

- Prevention of prescription errors like drug interactions (drug, food, disease, genes, etc.), Adverse Drug Events (ADE), human errors etc.
- Fewer callbacks from pharmacies: Poor physician handwriting can lead to pharmacists contacting the prescriber's office, seeking clarification about what exactly is being prescribed.
- Enable healthcare interoperability between various health entities • Reduce costs from printing paper prescriptions
- Quick access to and analysis of many prescriptions at once.
- High security due to the elimination of losing or forging prescriptions.
- Medication error rate reduction
- Saving health care costs like paper printing.
- Improving the quality of healthcare services and patient safety
- Facilitating the coordination of care and communication with other providers
- Cutting pharmacy costs through reduced redundancy
- Improving efficiency in the workflow of prescribers
- Speeding up prescription refill requests
- Facilitating the management of medication stocks
- Informing prescribers about patients' insurance coverage
- Improving patient convenience
- Checking patient adherence to medication to complete treatments
- Facilitating clinical decision-making by providing decision support features (such as checking for drug-drug, drug-allergy, and drug-dose interaction)
- Providing updated drug information like medication formularies
- Reducing wait times and increasing patient satisfaction with pharmacies

- Reducing the possibility that prescriptions are erased or lost

Despite the vast advantages brought by e-prescriptions, there are still some challenges that need to be addressed. Below are some of the challenges for e-prescriptions:

- Lack of standardized e-prescribing software
- Failure to properly implement e-prescribing
- Cost of e-prescribing implementation and maintenance
- Time required to integrate e-prescribing into the workflow
- Lack of computer support services
- Heterogeneous e-prescribing database management systems
- No drug information in the prescription
- Poor collaboration and no API for interoperability.
- Most e-prescription systems are not interoperable

Despite the advantages of semantic prescription, research on semantic prescription is at a snail's pace since current frameworks do not achieve these advantages especially in reducing prescription medication errors.

2.6.2 A review on Healthcare Interoperability ePrescription Frameworks

To develop quality healthcare interoperability eprescription systems, there is a need to review existing eprescription systems to determine the gaps and areas that need improvement. In this review, the eprescriptions are compared according to vital characteristics/metrics [41] Table 2-7 shows a comparative analysis of e-prescription frameworks in the literature. The majority of the frameworks do not vital eprescription features indicated with the “x” sign which shows a gap in the framework. A tick sign shows that there is no issues as the feature exists. The features are, Disease interaction, wrong drug, temperature monitoring, BMI monitoring, pulse-rate monitoring, blood pressure monitoring, SWRL inference, Conceptualization, excel import of axioms, Excel import of axioms, Sparql endpoint API, Public Ontology Publishing, Corpus-based ontology Evaluation, Competency Question, Evaluation using competency question, Evaluation for interoperability, Evaluation for usability, Ontology evaluation.

Table 2-7. A Comparison of eprescription Frameworks

	Framework Features	[42] O1	[43] O2	[44] O3	[45] O4	[46] O5	[47] O6	[48] O7	[49] O8	[50] O9	[51] O10	[52] O11	% of Gap in Literature	
1	Prototype Aspects	Has GUI	X	X	X	X	X	X	X	X	✓	✓	82%	
2		Open Source	X	X	X	X	X	X	X	X	X	X	100%	
3		No installation Required	X	X	X	X	X	X	X	X	X	X	100%	
4		Platform Independent	X	X	X	X	X	X	X	X	X	X	100%	
5		Secure	X	X	X	X	X	X	X	✓	✓	✓	✓	64%
6	Medical Errors	DDI Detection	X	X	X	X	X	X	✓	✓	✓	X	X	73%
7		Food interactions Detection	X	X	X	X	X	X	X	✓	X	✓	X	82%
8		Disease Interaction Detection	X	X	X	X	X	X	X	X	X	X	X	100%
9		Wrong Drug Error Detection	X	X	X	X	X	X	X	X	X	X	X	100%
10	Health Monitoring	Temperature Monitoring	X	X	X	X	X	X	X	X	X	X	X	100%
11		BMI Monitoring	X	X	X	X	X	X	X	X	X	X	X	100%
12			Pulse Rate Monitoring	X	X	X	X	X	X	X	X	X	X	100%
		Blood Pressure Monitoring	X	X	X	X	X	X	X	X	X	X	X	100%
13														
14	Semantic Aspects	Uses Ontologies	✓	✓	✓	✓	✓	✓	✓	X	✓	X	✓	82%
15		SPARQL Endpoint API	X	X	X	X	X	X	X	X	X	X	X	100%
16		Free Ontology Download	X	X	X	X	X	X	X	X	X	X	X	100%
17		SWRL Implementation	X	X	X	X	X	X	X	X	X	X	X	100%
18		Adhere to Formal Conceptualization Methodology	X	X	X	X	X	X	X	X	X	X	X	100%
19		Automated Axiom Import	X	X	X	X	X	X	X	X	X	X	X	100%
20	Ontology Evaluation	Formal Ontology Development	X	X	X	X	X	X	X	X	X	X	X	100%
21		Domain Knowledge Ontology Evaluation	X	X	X	X	X	X	X	X	X	X	X	100%
22		Competency Questions	X	X	X	X	X	X	X	X	X	X	X	100%
23		Interoperability Evaluation	X	X	X	X	X	X	X	X	X	X	X	100%
24		Usability Evaluation	X	X	X	X	X	X	X	X	X	X	X	100%
25		Ontology Pitfall Evaluation	X	X	X	X	X	X	X	X	X	X	100%	
26	Interoperability Aspects	Drug Information View	X	X	X	X	✓	✓	✓	X	X	✓	✓	55%
27		Patient access E-prescription history	X	X	X	X	X	X	X	✓	✓	✓	✓	64%
28		Pharmacist access Patient's history	X	X	X	X	X	X	X	X	X	X	X	100%
29		Prescriber access Patient's history	X	X	X	X	X	✓	✓	X	X	X	X	82%
30		Pharmacist Recommends Alternative Drug	X	X	X	X	X	X	X	X	X	X	X	100%
31		Prescription Statistical Analysis	X	X	X	X	X	X	X	X	X	X	X	100%
Percentage of Gaps per Framework		97%	97%	97%	97%	94%	90%	87%	87%	84%	84%	84%		

The author in [42] which is ontology 1(O1) proposed an e-Prescription ontology called PDRO with 14 domain concepts to solve the problem of inappropriate use of drug products. The author in [43] which is ontology2 (O2) created an ontology with 13 concepts for modeling drug-related knowledge and a repository of safe and effective generic prescription principles. The author in [53] is ontology3 (O3) illustrated the system architecture and an e-Prescription ontology with 13 domain concepts as well as the ontology specification languages that are required in implementing an e-Prescription system based on semantic web technologies. In [45], which is ontology4 (O4) the author proposed an e-Prescription ontology with 14 concepts to reduce the prescribing time and effort to select the desired drug. The author [46] which is ontology5 (O5) presented an e-Prescription ontology with 18 domain concepts to detect DDI and help patients, physicians, pharmacists, researchers, and pharmaceutical collaborate. The author in [54] which is ontology O6 proposed an e-Prescription ontology with 25 concepts for suggesting treatments for patients. In [48], the author proposed a pediatric e-Prescription ontology with 15 domain concepts to suggest the correct treatment for a patient

The problem with the above semantic e-Prescription frameworks is that most of them are still non-interoperable with other healthcare systems like the pharmacy, medical aid, insurance, patient or even future virtual pharmacies involving autonomous robotic dispensing [55] by machines and delivery by drones [56] to patient's houses. The e-Prescription systems are only accessed by the prescribers at the clinic or hospital and paper versions of the e-Prescription are printed and given to the patient to take to the pharmacy causing them to still be insecure as they can be lost or duplicated. To the best of our knowledge, the current frameworks do not allow the pharmacist to recommend alternative drugs to the patient when the specific drug is unavailable or too expensive by sending a real-time request to the prescriber system to validate the alternative drug and give a response with the updated prescription within seconds. This is very important in cases where the patient's health is in an emergency and moving from pharmacy to pharmacy might be detrimental. The current frameworks do not address prescription medical errors like DDI, drug-to-food interactions, drug-disease interactions, wrong drug, wrong dosage rate or strength etc. Currently, prescribers rely on drug interaction checking Application Programming Interfaces (APIs) like Drugbank, IBM

Micromedex, NLM, Truepill, IMgateway, PEPID, Cortellis etc. [57]. These APIs are very pricey and most of them cannot be integrated with the e-prescription system due to different terminologies and semantics, giving rise to the need for semantic frameworks. Prescribers are forced to leave the prescribing system Graphical User Interface (GUI) and check DDI manually. The author in [46] attempts to address DDI but does not show how DDI detection was implemented, and DDI detection performance metrics were not presented due to no results and evaluation shown. In addition, all these frameworks including [46] have a limited number of concepts which means they do not really cover the prescription domain knowledge and to the best of our knowledge all semantic e-Prescription frameworks were not evaluated or validated especially for domain knowledge coverage, as such, these ontologies cannot be trusted to adequately represent the e-Prescription domain. Ontology evaluation is a very crucial aspect of validating an ontology for quality and correctness and domain coverage [58]. Current e-prescriptions have no mechanism of helping pharmacists recommend an alternative drug for a patient by sending a request to the doctor for a prescription update if the drug is unavailable or if the drug is not covered by medical aid, and it's costly.

2.6.3 Summary of eprescription frameworks review

Figure 2-11 summarizes the percentage of gaps found in current eprescription interoperability frameworks. Figure 2-12 shows the gaps discovered in the current eprescription frameworks. This research will address all the gaps in Figure 2-12.

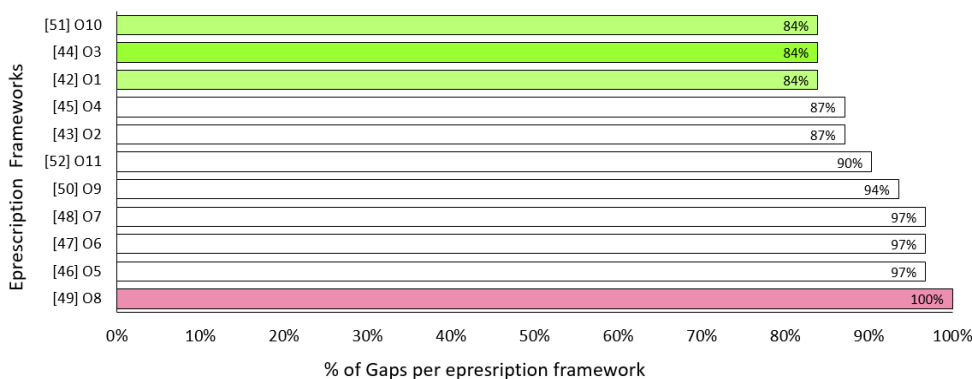
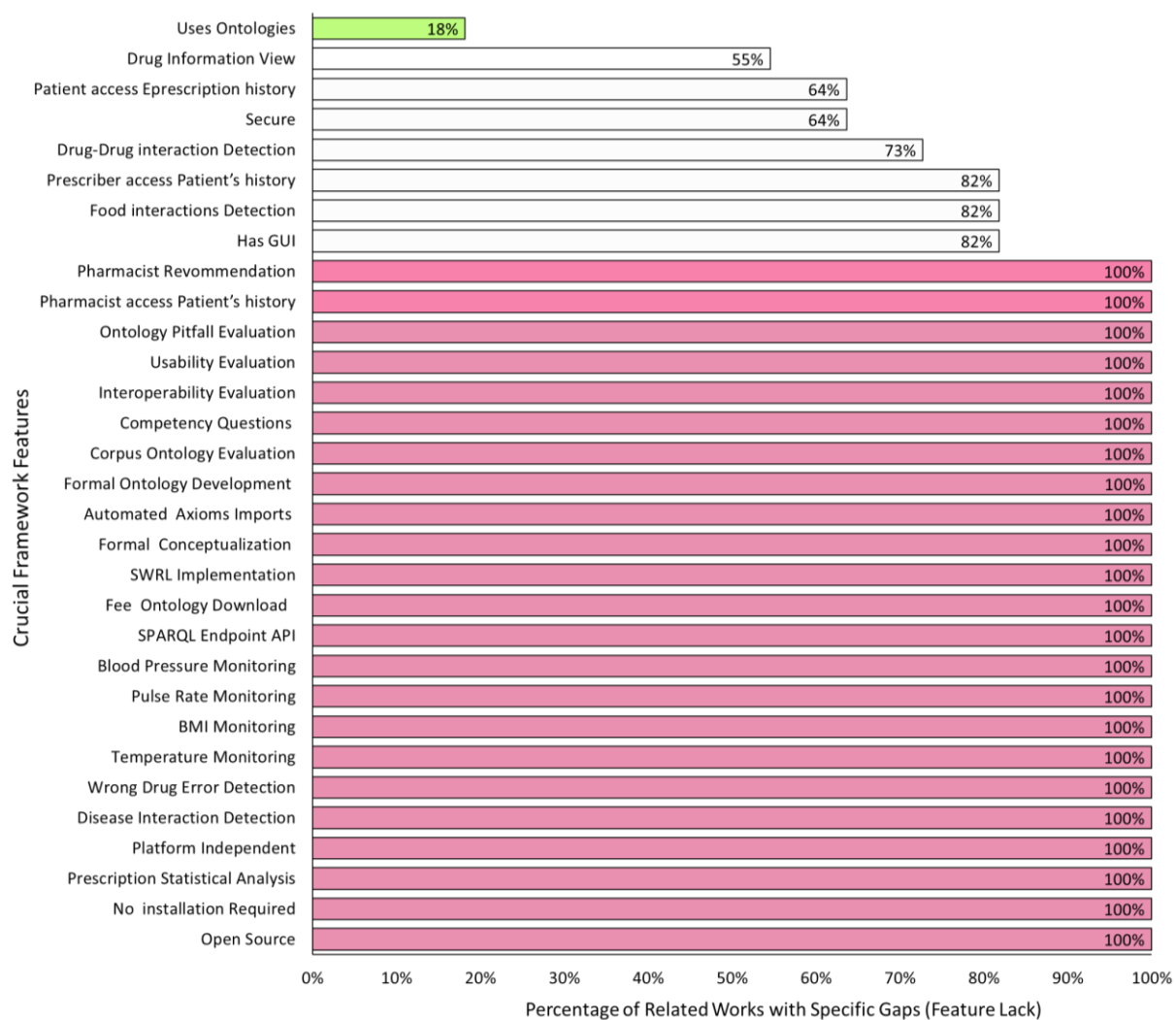


Figure 2-11 Percentage of gaps in each eprescription framework



CHAPTER 3 RESEARCH METHODOLOGY: PROPOSED IEPO FRAMEWORK

3.0 Introduction

This chapter presents the development of the proposed IEPO framework from the chosen and adopted ontology development methodologies, tools, and languages from the literature review with justification. The aim is broken down into five objectives in the introduction section. To accomplish the development of IEPO, all five objectives must be met. So far, the 1st and 2nd objectives below which deal with the literature review have been met in the literature review in Section.

Objective 1: To conduct a comprehensive literature review on semantic e-prescription interoperability frameworks to determine the existing gaps.

Objective 2: To conduct a comprehensive literature review to find the most suitable ontology development methodologies, tools, languages and most suitable ontology evaluation methodologies, tools and metrics to adopt for the proposed IEPO framework.

The 3rd and the 4th research objectives below have not been met and will be met in this chapter. In addition, these two objectives are the core for this implementation chapter as they deal with the implementation aspects of the IEPO framework.

Objective 3: To create two ontologies for both clinic and pharmacy health systems then merge them into one electronic prescription ontology called IEPO for healthcare interoperability.

Objective 4: To create a SPARQL endpoint Application Programming Interface (API) to enable various healthcare stakeholders like doctors, pharmacists, and patients to access IEPO healthcare data seamlessly and remotely via the internet.

The 5th, which is also the last research objective that deals with evaluating the proposed framework, will be achieved in the evaluation chapter. As presented in the introduction

chapter in Section 1.4, the main goal of this research is to develop an Intelligent e-prescription ontology for healthcare interoperability called IEPO.

From objective 3, two ontologies must be created for the clinic and pharmacy healthcare systems and then merged to form a single ontology called IEPO. The main goal of IEPO is to demonstrate healthcare interoperability between different non-interoperable healthcare systems. To narrow the scope of healthcare interoperability, which is very broad, the non-interoperable healthcare systems to be considered are the clinic and the pharmacy in terms of electronic prescription. For example, in countries like Botswana, private clinics and pharmacies are non-interoperable, and prescription information is not shared electronically between the two systems. Any two or more healthcare systems can be chosen to demonstrate healthcare interoperability but the justification for choosing the clinic and pharmacy systems is that they are the core for electronic prescription, which is the dominant method for achieving healthcare interoperability. It is assumed in this research that the clinic and pharmacy systems initially are noninteroperable, that is, they are unable to share information and with other healthcare systems entities like the patient electronically and IEPO framework will be the bridge to make the systems interoperate and share data. The process is to create the clinic ontology and pharmacy ontology at the conceptualization level and merge the two semi formal ontologies form IEPO.owl ontology file, host the IEPO.owl ontology file on a triple store server online, and query the triple store from a SPARQL Endpoint API to disseminate healthcare systems information to any authorize healthcare system. With IEPO, any authorized healthcare entity can access specific healthcare information from other systems via the SPARQL endpoint API remotely. In order to accomplish objective 3, there is a need for answers to research questions 1 to 5 from the literature review in Section 2.5 which give justification of the choice of ontology development and evaluation methodologies, languages, tools, and metrics from literature suitable for IEPO framework development. Below are the answers to research question 1 to 5 with section references that points to the justification statements in the literature review section.

Research Question 1: What are the gaps found in current healthcare interoperability frameworks from literature based on semantic eprescription to be addressed by the proposed IEPO framework?

Gaps to be addressed by IEPO: The answer is presented in Section 2.6.3. Many gaps have been identified in the literature and summarized in Section 2.6.3 Figure 2-11 and Figure 2-12 with the percentage of current frameworks presenting the specific gap. All gaps identified in the literature shown in Table 3-1 will be addressed in this research. The aim is to address gaps not addressed in the literature and those in more than 50% of the literature, as shown in Table 3-1). Since all the specific gaps exist in more than 50% of related works except one, which is 18% (use of ontologies), all the gaps shown in Table 3-1 will be addressed by IEPO;

Table 3-1 Gaps to be addressed in this research

	Identified Gaps		% of the Gap in Literature	IEPO Framework Addressed (✓), Not Addressed (X)
1	Prototype Aspects	Has GUI	82%	✓
2		Open Source	100%	✓
3		No installation Required	100%	✓
4		Statistical Analysis	100%	✓
5		Platform Independent	100%	✓
6		Secure	64%	✓
7	Medical Errors	DDI Detection	73%	✓
8		Food interactions Detection	82%	✓
9		Disease Interaction Detection	100%	✓
10		Wrong Drug Error Detection	100%	✓
11	Health Monitoring	Temperature Monitoring	100%	✓
12		BMI Monitoring	100%	✓
13		Pulse Rate Monitoring	100%	✓
14		Blood Pressure Monitoring	100%	✓
16	Semantic Aspects	Uses Ontologies	82%	✓
17		SPARQL Endpoint API	100%	✓
18		Free Ontology Download	100%	✓
19		SWRL Implementation	100%	✓
19		Formal Conceptualization Methodology	100%	✓
20		Automated Axioms Imports	100%	✓
21		Formal Ontology Development	100%	✓
22	Ontology	Domain Knowledge Ontology Evaluation	100%	✓

23	Evaluation	Competency Questions	100%	✓
24		Interoperability Evaluation	100%	✓
25		Usability Evaluation	100%	✓
26		Ontology Pitfall Evaluation	100%	✓
27	Interoperability Aspects	Drug Information View	55%	✓
28		Patient access E-prescription history	64%	✓
29		Pharmacist access Patient's history	100%	✓
30		Prescriber access Patient's history	82%	✓
31		Pharmacist Recommends Alternative Drug	100%	✓
		Prescription Statistical Analysis	100%	✓

Research Question 2: What are the most suitable healthcare interoperability types and approaches to adopt for developing the IEPO framework?

- Adopted Interoperability Type: Semantic Interoperability with justification in the introduction section
- Adopted Interoperability Approach: Centralized Approach with justification in the introduction section

Research Question 3: What are the most suitable interoperability standards to adopt for developing the IEPO framework?

- Adopted Healthcare Interoperability Standard: SNOMED with justification in the introduction section

Research Question 4: Between ontologies and databases, which is the most suitable to adopt for developing an eprescription interoperability framework?

- Ontologies with justification in the literature review section

Research Question 5: What are the most suitable ontology development methodologies, languages, and tools to adopt for the development of IEPO framework?

- Adopted Ontology Development Methodology from Literature Review: Methontology with justification in the literature review section
- Adopted Ontology Development Languages from Literature Review: OWL2 with justification in the literature review section.
- Adopted Ontology Development Tools from Literature Review: Protégé 5.5.0 with justification in the literature review section

The other research questions deals with the implementations and will be covered by this chapter.

Other tools were adopted for ontology hosting but were not in the literature review due to limited space.

This chapter will describe how Protégé was used to develop the IEPO ontology framework.

Description scenario of the case study between the clinic and pharmacy healthcare systems:

There exist two stand-alone and non-interoperable healthcare systems; the clinic and the pharmacy healthcare systems. The clinic healthcare system comprises the following major concepts; doctor, clinic, patient, receptionist, patient's prescription, drugs within patient's prescription, patient's disease, patient's appointment, food taken by the patient, patient allergies, etc.

The pharmacy healthcare system comprises the following major concepts; pharmacist, pharmacy, patient, patient's prescription, drugs within patient's prescription, patient's disease, food taken by the patient, patient allergies, etc. It is assumed that all pharmacies, clinics, doctors, and patients have all their details registered with The IEPO framework.

A brief summary on addressing the gaps in literature: The gaps in Table 3-1 will be addressed by the IEPO framework as follows; Interoperability will be addressed using ontologies through a web-based prototype and an Application Programming Interface (API). The ontology will be developed following formal ontology development and conceptualization methodologies adopted from the literature. The barriers with manual and time-consuming axiom creations in ontology development tools like Protege will be overcome using automated axiom imports plugins using Cellfie, A simple web-based IEPO prototype will be developed to demonstrate interoperability between the doctor, pharmacist, and patient. A patient with a disease sets up an appointment with a doctor at a particular clinic. The doctor diagnoses the patient with a particular sickness or disease based on some symptoms and tests and prescribes some drugs using the developed IEPO prototype's Graphical User Interface (GUI). Because of the web-based nature of the IEPO framework, it requires no installation, making it platform-independent. The security gap is addressed by access control using passwords. Patients, doctors, and pharmacists will log in to IEPO GUI using a username and password. In addition, the IEPO is made open

source, with the developed IEPO ontology made freely available for download for further research and large-scale deployment by healthcare organizations and the government.

The API will be a web-based SPARQL Endpoint where competency question queries are performed remotely by a patient, doctor, or pharmacist on the IEPO ontology containing data of all the healthcare systems. Interoperability is achieved when patients, pharmacists, and prescribers (doctors) can seamlessly and remotely access the patient's prescription data and medical history. The IEPO framework also performs statistical analytics on the prescription data and give results of queries like "The most prescribed drug between January and April 2022 in all clinics", "The most diagnosed disease in the year 2022 in all patient," or "The total number of prescription prescribed in the year 2022 by a particular doctor" etc. In addition, the IEPO framework performs some medical error checking using Semantic Web Rule Language (SWRL) and gives alerts during prescribing. With IEPO, medical error checking is incorporated with the prescribing platform such that when there is a prescription error, an alert is triggered before sending the prescription to the pharmacy of the patient's choice. The medical prescription errors considered are; DDI, food interactions, disease interactions, and the prescription of a wrong drug for a particular disease. The patient's prescription data and history can be accessed by all the registered and authorized doctors in all the clinics. The accessibility of all patient's prescription data by doctors means any registered doctor can view the history of all the patients they attend to, which can avoid double prescriptions by doctors and drug interactions.

The IEPO framework also performs healthcare monitoring of vital signs via SPARQL language and gives alerts when abnormalities are detected. The vital signs considered are; temperature, BMI, Pulse rate, and blood pressure. With IEPO, as soon as the patient is taken body temperature during an appointment with the doctor, the temperature value is sent to the IEPO framework, which can detect abnormalities like Covid19 temperature ranges and give alerts immediately. The same is applied to abnormalities ranges for BMI, pulse rate, and blood pressure. The status of these vital signs is registered permanently in the IEPO framework as medical history within the [patient's electronic prescription. The doctor then submits the eprescription to the central IEPO server to be stored permanently. The patient then goes to the pharmacy of their choice to have the prescription dispensed, where the pharmacist will perform identity checks on the patient and then search the web base prototype or query the SPARQL Endpoint using the patient's identification number (ID) or prescription ID. Suppose the

prescription drug is not available or is costly. In that case, the pharmacist recommends an alternative, cheaper drug for the same sickness condition by sending a prescription update request to the doctor to validate by approving or rejecting the pharmacist's request. If not approved, the patient can go to the following pharmacy. If approved, the medication is then dispensed and marked as dispensed, and a refill date, if any, is triggered. The patient can then access prescriptions history on the IEPO GUI.

3.1 Ontology Development for the Proposed IEPO Framework

Figure 3.1 shows the adopted ontology development methodology called Methontology from [60][28][61][29], with the description of each phase on the right side inside the figure. All the 12 Methontology activities in Figure 3.1 were carried out and documented in very large tables and complex diagrams, for medication using the IEPO GUI. The reader is directed to [61][29] and [62][28] for more elaborate descriptions of Methontology activities.

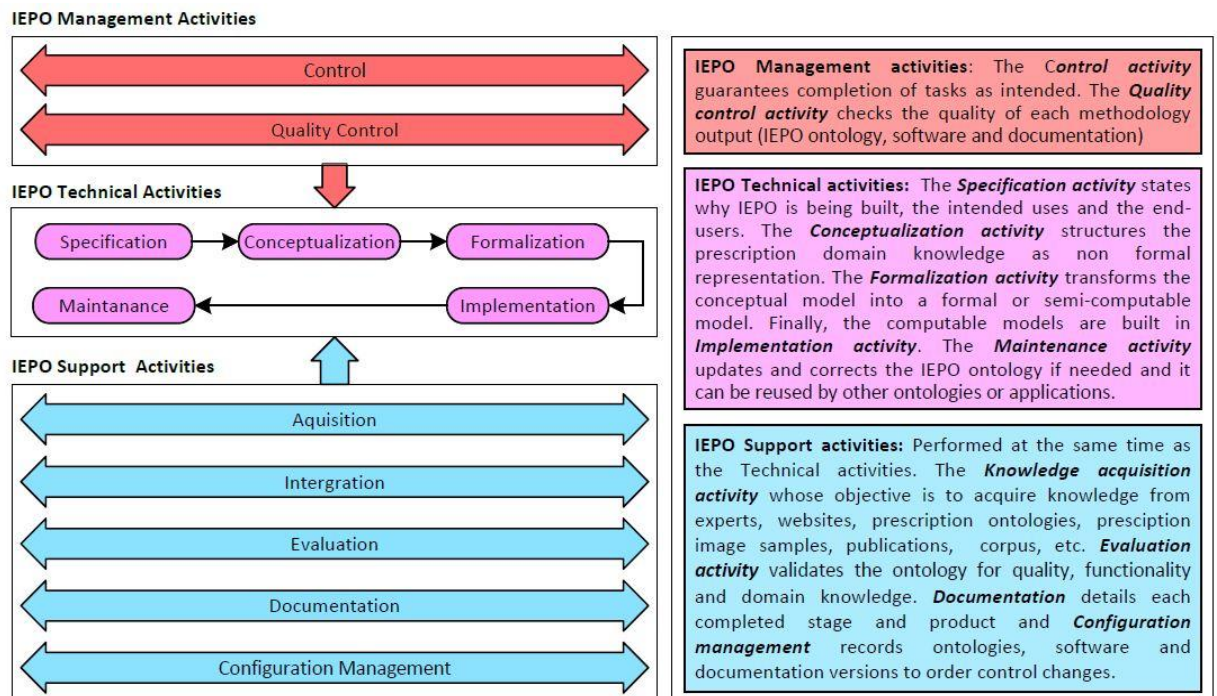


Figure 3-1 Adopted Ontology Development Methodology (Methontology) for the proposed IEPO Framework

3.2 Chapter Terminologies.

Below are some main terminologies used in this research work.

- Concepts/classes are mainly the classes and subclasses describing an ontology like Prescription or Patient.
- Relations represent a type of association between concepts of the domain like HasPrescription.
- Linked data
- Triple
- SPARQL Endpoint
- Subject/Domain
- Object/Range
- A predicate is relation that connects the subject with the object. A predicate can either be an object or data property.
- Instances or Individuals are used to represent elements or individuals in an ontology like the sickness of Schizophrenia. Instances will be used interchangeably with individuals.
- Object properties connect two individuals or classes (a subject and object) with a predicate.
- Data properties – is when the predicate connects a single subject with some attribute data. Data properties have defined datatypes, including string, integer, date, DateTime, or boolean.
- Attributes describe properties of instances and concepts like Firstname or PatientID.
- Rules are generally used to infer knowledge in an ontology like SWRL.
- Axioms are logical expressions that are always true and are typically used to specify constraints in the ontology, like the statement Patient Subclass Person.

3.3 Acquisition Activity

Figure 3.2 shows the Acquisition Activity that was carried out through data collection from ontologies about (prescription, drug information and drug interactions), websites, sample

image prescriptions from google, e-prescription corpus, publications, and general knowledge. The acquisition activity enabled e-prescription concepts extraction common in the prescription domain from various sources.

3.4 Specification Activity

Figure 3.2 shows the Specification Activity of IEPO which entails the ontology scope. The scope of IEPO is to demonstrate healthcare interoperability and prevent medical prescription errors by sharing e-prescription information between different healthcare entities like a doctor, pharmacists, patients, medical aid, insurance, research, and machines while relying on up-to-date drug information sources. The specification was also incorporated with the requirements of IEPO which are a set of competency questions the ontology is supposed to answer. There are many competency questions derived but a few are listed as follows:

1. What are all the prescriptions for patient A between date A and date B?
2. What are all the current sicknesses for patient A and the prescription prescribed for each sickness?
3. What are all the prescriptions prescribed by doctor A? between date A and date B?
4. Is prescription A for patient B dispensed yet? If yes when, where and by who?
5. How many prescriptions have been prescribed from pharmacy A from date B to date C for treating disease D?
6. Which is the most prescribed drug in all pharmacies between date A and date B?
7. Will drug A interact with drug B?
8. Will the current drug A to be prescribed for patient B interact with all the current medications/drugs patient B is taking?
9. Is drug A currently prescribed, the correct drug for sickness B diagnosed?
10. What are the alternative drugs for a particular sickness A and their prices?
11. To Doctor A, can I dispense as pharmacist B an alternative drug C in place of Drug D to patient E due to the high cost or shortage of drug D?
12. Given drug A and Drug B which both treat sickness C, which of the two drugs is cheaper and has no interaction with current patient medication?

13. Are there any DDIs or wrong drug errors in the current patient's prescriptions?

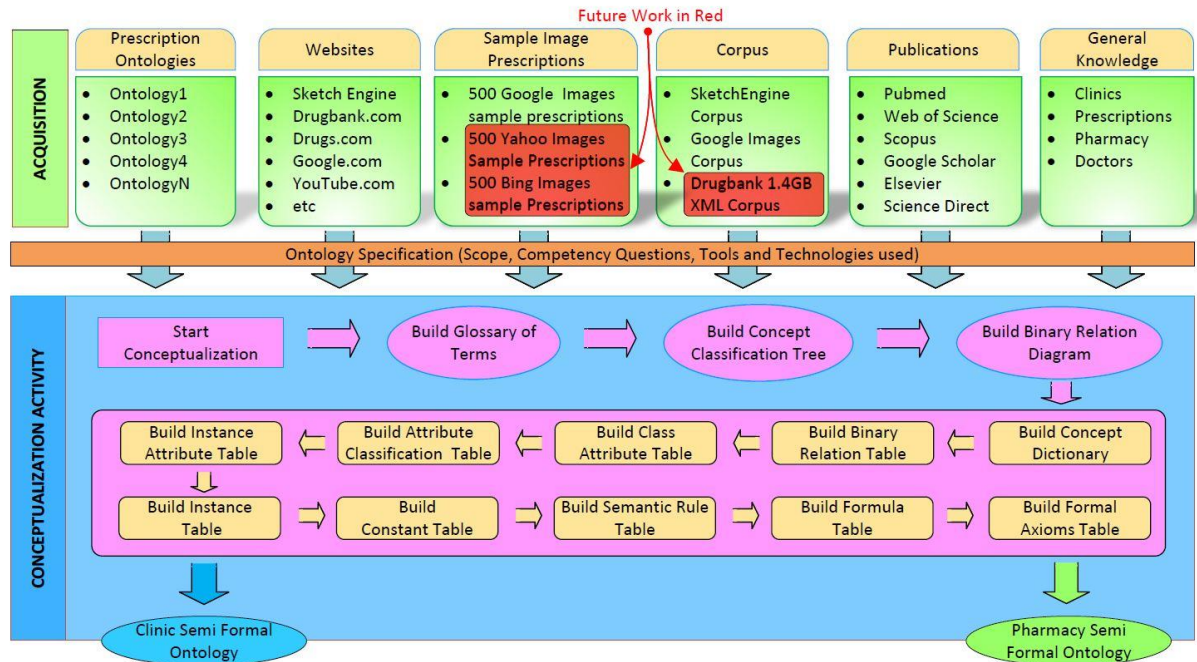


Figure 3-2 IEPO Acquisition, Specification and Conceptualization Activities

The ontology specification also captures the tools and technologies to be used which are not limited to Protégé [63] for implementation, Jena Fuseki server for hosting IEPO as a triplestore database, Cross-Platform Apache, MySQL, PHP, and Perl (XAMPP) server, PHP server to host PHP prescribing software prototype, SPARQ [64] which is a recursive acronym for SPARQL Protocol and Resource Description Framework (RDF) Query Language, Representational State Transfer (REST) protocol, Web Ontology Language (OWL), etc. These tools and technologies will be described later in this research work.

3.5 Conceptualization

Conceptualization is a semi-formal way of representing an ontology before implementation into a formal one as explained previously in Figure 3.1. Figure 3.2 shows 13 conceptualization sub-activities from building a glossary of terms to formal axioms table. These activities were created for both the clinic and pharmacy making a total of 26 activities. All 26 activities were

carried out following the work from [65] and [66]. These two publications give very detailed and easy-to-follow conceptualization activities with examples as such, the reader is directed to [65] and [66] for more detailed conceptualization activity explanations. Not all of the conceptualization activities will be displayed but a few due to space constraints.

3.5.1 Building the Glossary of Terms

3.5.1.1 *Glossary of Terms of Names and Concepts*

Table 3.1 shows the glossary of terms which consists of the names of the concepts of the proposed IEPO ontology and their descriptions.

Table 3-2 below shows the proposed IEPO ontology of the proposed research work for the Clinic and pharmacy which shows the names (Drug, Prescription, person, Patient, pharmacy, Clinic, Disease, Doctor, allergy, Receptionist, Pharmacist, and Food) of the concepts and their description.

Table 3-2 Glossary of terms of the proposed IEPO Ontology

Name	Description
Drug	A medical substance that is prescribed by a doctor to a patient.
Prescription	A certified document written by a doctor that authorizes a patient to be issued with a drug for disease treatment.
Person	A human being
Patient	A person receiving medical treatment.
Pharmacy	A medical place that dispenses a prescription to a patient.
Clinic	A health care institution where patients receive medical treatment
Disease	A definite pathological process having a characteristic set of signs and symptoms
Doctor	A medical practitioner offers patients medical treatment and prescriptions.

Allergy	A reaction or anomaly that is being diagnosed in a patient by a doctor
Receptionist	A person who registers patients at a clinic to see a doctor.
Pharmacist	A person licensed to prepare and dispense drugs.
Food	A substance that is taken by a person to satisfy one's stomach

3.5.1.2 Class Object Properties Glossary of Terms

Table 3-3 shows the glossary of the term of a class object which contains the class domain (allergy, appointment, patient, prescription, pharmacist, doctor, etc.) object property, class range, and description of the IEPO ontology.

Table 3-3 Class Object Properties Glossary of Terms

Class Domain	ObjectProperty	Class Range	Description
Allergy	belongs to	Patient	An allergy belongs to a patient, a disease belongs to a patient, a prescription belongs to a patient, a drug belongs to a prescription, and a subscription belongs to a prescription.
Appointment	BookedBy	Patient	An appointment is booked by a patient,
Patient	Books	Appointment	
Prescription	DispensedBy	Pharmacist	Prescription is dispensed by a pharmacist; a pharmacist dispense prescription.
Pharmacist	Dispenses	Prescription	
Doctor	EmployedBy	Clinic	A doctor is employed by a clinic. Pharmacist is being employed by a pharmacy.
Pharmacist	EmployedBy	Pharmacy	
Receptionist	EmployedBy	Clinic	Receptionist is employed by a clinic, a clinic employs a receptionist, a clinic employs a doctor, A pharmacy is employs by a pharmacist.
Clinic	Employs	Receptionist	
Clinic	Employs	Doctor	
Pharmacy	Employs	Pharmacist	
Patient	HasAllergy	Allergy	A patient has an allergy.
Patient	HasDisease	Disease	A patient has a disease, a disease has a disease interaction drug, a drug has a disease interaction disease.
Disease	HasDiseaseInteraction	Drug	
Drug	HasDiseaseInteraction	Disease	
Patient	HasDoctor	Doctor	

Prescription	HasDrug	Drug	A patient has a doctor, a prescription has a drug drug, a subprescription has drug drug.
SubPrescription	HasDrug	Drug	
Drug	HasDrugInteraction	Drug	A drug has drug interaction drug, a drug has food interaction food.
Drug	HasFoodInteraction	Food	
Disease	HasIndication	Drug	A disease has an indication of drug,
Doctor	HasPatient	Patient	a doctor has a patient.
Patient	HasPrescription	Prescription	a patient has a prescription
Prescription	HasSubPrescription	SubPrescription	, a prescription has a sub prescription sub prescription,
Prescription	PrescribedBy	Doctor	, a prescription is prescribed by a doctor, a doctor prescribes prescription
Doctor	Prescribes	Prescription	
Patient	RegisteredBy	Receptionist	A patient is register by a receptionist, a receptionist registers by a patient
Receptionist	Registers	Patient	
Food	TakenBy	Patient	A food is taken by a patient, a patient takes a food,
Patient	Takes	Food	
Drug	Treats	Disease	a drug treats a particular disease.

3.5.1.3 Prescription Class Glossary of Terms

The prescription class glossary of terms is shown in **Table 3-4** It contains the data property (Prescription, Quantity, NextFill, DosageForm, DosageRoute, DosageStrength, Date, PrescriptionID, DispensedStatus, DDIStatus, FoodInteractionStatus, DiseaseInteractionStatus, WrongDrugStatus, InstructionQuantity, InstructionFrequency, InstructionDescription, Price) and the description of the proposed IEPO ontology.

Table 3-4 Prescription Class Glossary of Terms

Data Property	Description
Prescription	An instruction written by a medical practitioner that authorizes a patient to be issued with a medicine or treatment
Quantity	The Number of drugs that is being prescribed by the doctor.
NextFill	The number of days until the next prescription is repeated.

DosageForm	Is the physical form of a dose of a chemical compound used as a drug or medication intended for administration or consumption. Eg pill, tablet, capsule etc.
DosageRoute	Is the way by which a drug fluid or other substance is taken into the body.
DosageStrength	The proportion of active drug substances to expedient, measured in units of volume or concentration.
Date	The time in which the prescription is assigned.
PrescriptionID	The instructionID of the drug that is prescribed to the patient by the doctor eg PA00000001
DispensedStatus	This is the status of the drug that is dispensed by the pharmacist.
DDIStatus	This is the interaction of the two medical substance that is prescribed by the medical practitioners to the patient.
FoodInteractionStatus	This is the meal that interacts with a drug that is being prescribed by the doctor
DiseaseInteractionStatus	The status of the disease-to-disease interaction if it exists in the prescription.
WrongDrugStatus	This shows the status if there is an alert in the wrong prescription by the doctor for a particular wrong disease
InstructionQuantity	This shows the number of drugs that is in the prescription of the patient, prescribed by the doctor.
InstructionFrequency	This shows how often a prescribed drug is to be taken by the patient which was recommended by the doctors
InstructionDescription	This narrates how the drug is being prescribed by the doctor
Price	This is the amount that is to be paid for each unit of the prescribed drug which is being prescribed by the doctor and dispensed by the pharmacist.

3.5.1.4 Patient Class Data Properties Glossary of Terms

Table 3-5 below shows the Patient class data properties glossary of terms which consist of data property (PatientID, Address, DOB, Age, Gender, phone, FirstName, LastName, BloodPressure, PulseRate, Weight, Temperature, Height and BodyMassIndex) and the description of the proposed IEPO ontology.

Table 3-5 Patient Class Data Properties Glossary of Terms

Data Property	Description
PatientID	The unique identity of each patient
Address	The location of a place where a person lives.
DOB	The year, month and day a person was born.
Age	The number of years from the time the person was born
Gender	Either of the two sexes (male and female)
Phone	A person's mobile number
FirstName	A person's name before the last name.
LastName	A person's surname or family name
Blood Pressure	The blood pressure in the blood vessels in millimetres of mercury (mmHg)
PulseRate	Patient's Heart rate or rhythm: the number of times the heart beats per minute (BPM)
Weight	The mass of a patient in kg.
Temperature	This is the patient's temperature in °C
Height	The tallness of a patient in meters
BodyMassIndex	The ratio of the patient's weight to the square of height used to categorize the weight as underweight, normal weight, or overweight measured in kg/m2

3.5.1.5 Drug Concept Data Properties Glossary of Terms

The drug concept data properties glossary of terms for the proposed IEPO ontology and their description are shown in **Table 3-6**.

Table 3-6 Drug Concept Data Properties Glossary of Terms

Drug Data Property	Description
SNOMEDCODE	This is the standard code for medicine which is used in identifying a particular drug in the domain.
SideEffect	The damage or danger of taking the wrong drug by a patient.
RouteOfElimination	This is the removal of an administered drug from the body. It is accomplished in two ways, either by excretion of an unmetabolized drug in its intact form

	or by metabolic biotransformation followed by excretion removal of drug from the body.
ProteinBinding	This is the complex formation of drug with protein. The proteins are particularly responsible for such an interaction
Metabolism	This describe the biotransformation of pharmaceutical substances in the body so they can be eliminated more easily.
MechanicOfAction	This describes how a drug or other substance produce an effect in the body
DrugBankID	This is the ID of a particular drug in the drugbank
HaLfLife	This is the time it takes for drug active substance in your body to reduce by half.
Drugname	The name of the prescribed drug by the doctor
Absorption	It refers to the way a drug is absorbed from a pharmaceutical formulation into the bloodstream.
DosageRoute	This is how the prescribed drug is to be taken by the patient
DosageStrength	The proportion of active drug substance to excipient, measured in units of volume or concentration.
Alternative	The suggestion of a particular drug in the absence or unavailability of the prescribed drugs
AvearageMass	An amount of drug in unit of mass.
Brand	This is another form of the prescribed drug.
UnitPrice	The amount to be paid for a single amount of a drug
DosageForm	This is the form of the medical substance of the dose that is being prescribed by the doctor.

3.5.1.6 Clinic and Pharmacy Concept Data Properties Glossary of Terms.

Table 3-7 shows the Clinic and Pharmacy Concept Glossary of Terms of the proposed IEPO ontology which consists of the Class Attribute (Name, Address, Phone) and the description of each of the Class Attributes.

Table 3-7 Clinic and Pharmacy Properties Glossary of Terms

Class Attribute	Description
Name	The name of the clinic or pharmacy
Address	The physical address of the clinic or pharmacy
Phone	The clinic or pharmacy healthcare system phone number

3.5.1.7 Disease Concept Data Properties Glossary of Terms

Table 3-8 shows the disease concept data properties glossary of terms of the IEPO ontology, and it consists of the Class Attribute (Name, Symptoms) and the description of each attribute.

Table 3-8 Disease Concept Data Properties Glossary of Terms

Class Attribute	Description
Name	The name of a patient's sickness or disease.
Symptom	The disease symptoms

3.5.1.8 Doctor/Pharmacist/Receptionist Concept Data Properties Glossary of Terms

Table 3-9 below shows the doctor and pharmacy concept data properties glossary of terms of the IEPO ontology, it consists of the class attribute which includes the FirstName, LastName and also the description of each of the concepts.

Table 3-9 Doctor/Pharmacist/Receptionist Concept Data Properties Glossary of Terms

Class Attribute	Description
FirstName	The first name of a Person
LastName	The surname of the Person

3.5.1.9 Food\Allergy Concept Data Properties Glossary of Terms

Table 3-10 below shows the food and allergy concepts data properties glossary of terms which include the class attribute and description are shown in Table 3.19.

Table 3-10 Food/Allergy Concept Data Properties Glossary of Terms

Class Attribute	Description
Name	The name of the food or the allergy

3.5.2 Build the concept-classification trees:

The concept-classification trees consist of relations such as subclass-of, sub-class-partition-of, and exhaustive-subclass-of. (Class C is a subclass of parent class P if and only if every instance of C is also an instance of P. A subclass partition of C is a set of mutually disjoint subclasses of C. An exhaustive subclass of C is a set of mutually- disjoint classes—a subclass partition—that covers C. Every instance of C is an instance of exactly one of the subclasses in the partition.) so the proposed domain taxonomies will form the proposed ontology healthcare systems which consist of the Clinic and pharmacy ontologies as shown in Figures 3.4 and 3.5 respectively.

3.5.2.1 Clinic Concept Classification Tree

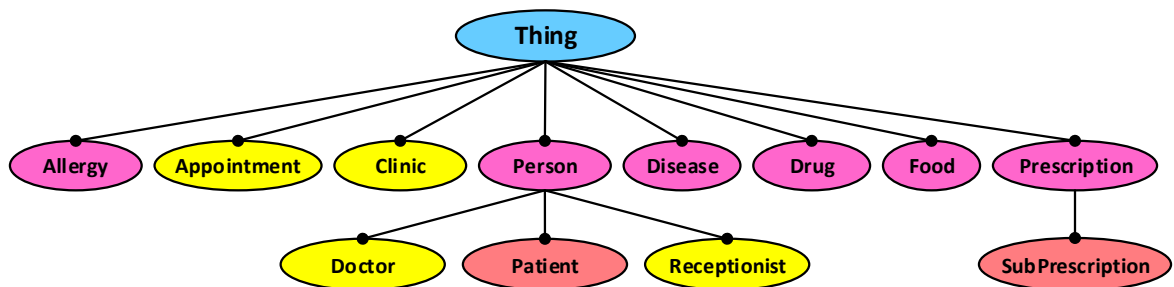


Figure 3-3 A Concept Classification Tree for the Clinic

3.5.2.2 Pharmacy Concept Classification Tree

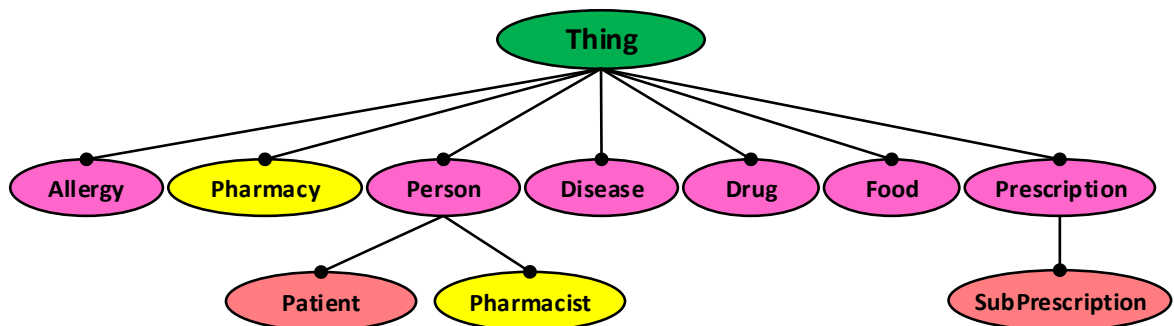


Figure 3-4 A Concept Classification Tree for the Pharmacy

3.5.3 Binary Relations Diagram

A binary-relations table specifies the name, the names of the source and target concept, the inverse relation, and so on for each ad hoc relation whose source concept is in the concept-classification trees. Each binary concept links the concepts in the opposite direction. For example, a patient books an appointment and the inverse is an appointment booked by a patient, so it links the concepts and creates their inverse.

—Clinic Binary Relation Diagram

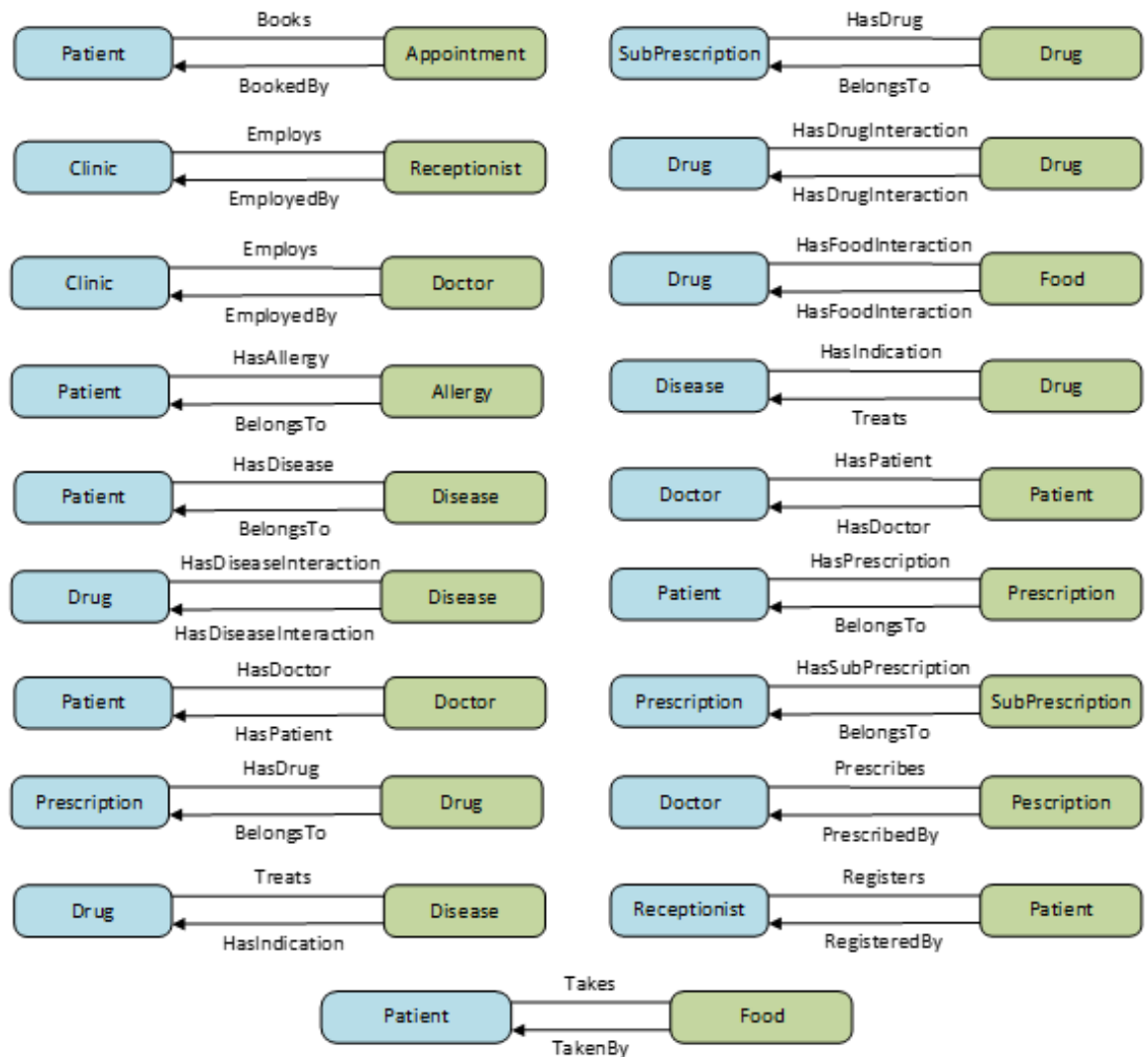


Figure 3-5 Clinic Binary Relation Diagram

3.5.3.1 Pharmacy Binary Relation Diagram

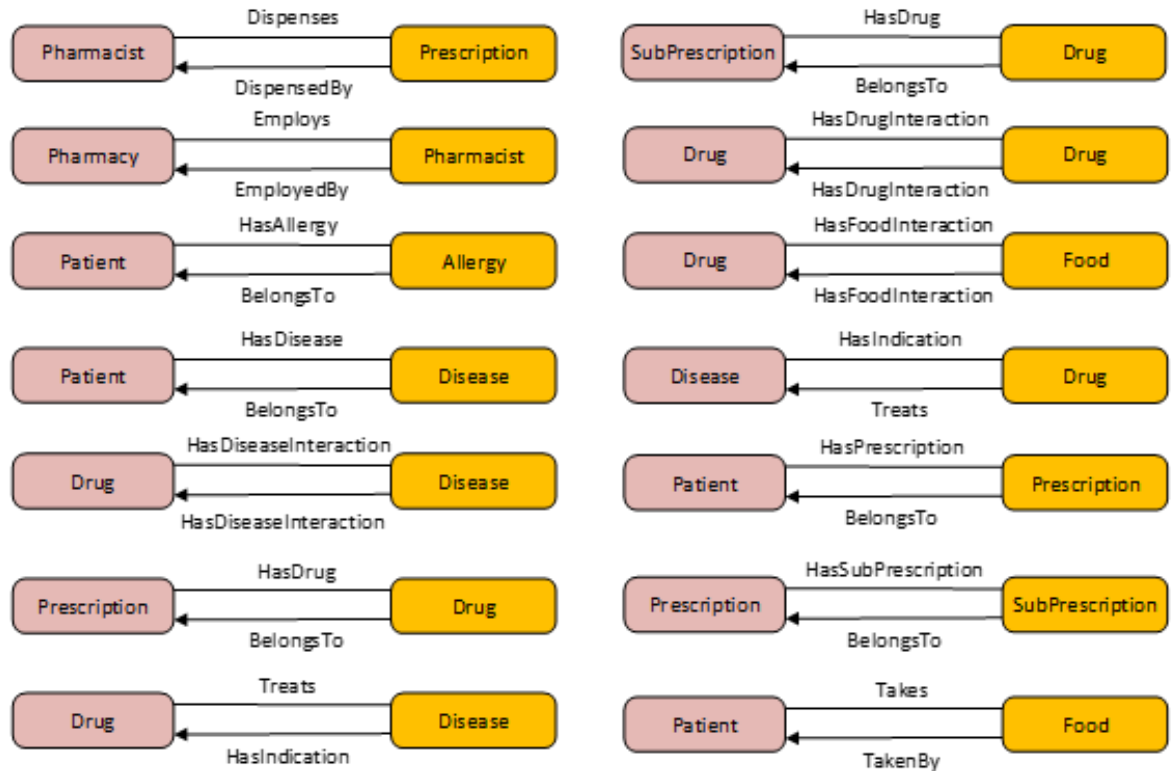


Figure 3-6 Pharmacy Binary Relation Diagram

3.5.4 Concept Dictionary

Table 3-11 below shows the *concept dictionary* containing all the domain concepts, instances of such concepts, class, and instance attributes of the concepts, and, optionally, concept synonyms and acronyms give a small part of the concept dictionary of the clinic and pharmacy healthcare system.

Table 3-11 Concept Dictionary

Concept	Instances	Instance Relation	Instance attribute
Drug	Risperidone Ritonavir Quetiapina Metoclopramide	Treat Belongs HasDrugInteraction HasFoodInteraction DiseaseInteraction	Description Toxicity SnomedCode SideEffect RouteOfElimination

			ProteinBinding Metabolism MechanismOfAction HalfLife DrugName Absorption DosageRoute DosageStrength Alternative AverageMass Brand UnitPrice DosageForm DrugBankID
Prescription	JedaJoelPR00000001 JedaJoelPR00000002 AbrahamJamesPR00000003S P1 AbrahamJamesPR00000003S P2 DorcasHopePR00000004 AbelJohnPR00000005 AbrahamJamesPR00000003	PrescribedBy BelongsTo HasDrug DispensedBy	PrescriptionID DispensedStatus DDIStatus FoodInteractionStatus DiseaseInteractionStatus WrongDrugStatus InstructionQuantity InstructionFrequency InstructionDescription Price
Person	None (Patient, Pharmacist, Doctor, and receptionist instances are inherited from Person Class as persons)	None	Address DOB Age Gender Phone Address Firstname Lastname
Patient	JedaJoel AbrahamJames DorcasHope AbelJohn	HasDoctor HasDisease HasPrescription Books RegisteredBy HasAllergy Takes	Address DOB Age Gender Phone Address Firstname Lastname BloodPressure PulseRate Weight Allergy Temperature Height BodyMassIndex

			PatientID
Pharmacy	GodwinBathoPharmacy	Employs	Name Address Phone
Clinic	DavidSamClinic JimMayerClinic	Employs	Name Address Phone
Disease	Schizophrenia GastroesophagealReflux HIV-1 Dementia	BelongsTo HasIndication HasDiseaseInteraction	Name Symptom
Doctor	DavidSam JimMayer	EmployedBy Prescribes HasPatient	(Inherits all Person Concept instance attributes)
Allergy	Alcohol Beef Milk Rice Non	BelongsTo	Name
Receptionist	WilsonFrancis MendisAlahambra	Registers EmployedBy	(Inherits all Person Concept instance attributes)
Pharmacist	GodwinBatho	Dispenses EmployedBy	(Inherits all Person Concept instance attributes)
Food	Alcohol Milk Yogurt Cheese		Name
Appointment	AP1 AP2 AP3 AP4 AP5	BookedBy	Date Type

3.5.5 A Binary-Relation Table

Table 3.11 below shows the binary relation table of the IEPO ontology which consists of the relation Name example as follows the books, employs, HasAllergy, HasDisease, HasDiseaseinteraction, HasDoctor, HasDrug, HasDrugInteraction, HasFoodInteraction, HasIndication, HasPatient, HasPrescription

Table 3-12 Binary-Relation Table of the IEPO Ontology

Relation Name	Source Concept	Target Concept	Target Cardinality	Inverse Relation
Books	Patient	Appointment	{1, M}	BookedBy
Dispenses	Pharmacist	Prescription	{1, M}	DispensedBy
Employs	Clinic	Receptionist	{1, M}	EmployedBy
Employs	Clinic	Doctor	{1, M}	EmployedBy
Employs	Pharmacy	Pharmacist	{1, M}	EmployedBy
HasAllergy	Patient	Allergy	{1, M}	BelongsTo
HasDisease	Patient	Disease	{1, M}	BelongsTo
HasDiseaseInteraction	Drug	Disease	{1, M}	HasDiseaseInteraction
HasDoctor	Patient	Doctor	{1, M}	HasPatient
HasDrug	Prescription	Drug	{1, M}	BelongsTo
HasDrug	SubPrescription	Drug	{1, 1}	BelongsTo
HasDrugInteraction	Drug	Drug	{1, M}	HasDrugInteraction
HasFoodInteraction	Drug	Food	{1, M}	HasFoodInteraction
HasIndication	Disease	Drug	{1, M}	Treats
HasPatient	Doctor	Patient	{1, M}	HasDoctor
HasPrescription	Patient	Prescription	{1, M}	BelongsTo
HasSubPrescription	Prescription	SubPrescription	{1, M}	BelongsTo
Prescribes	Doctor	Prescription	{1, M}	PrescribedBy
Registers	Receptionist	Patient	{1, M}	RegisteredBy
Takes	Patient	Food	{1, M}	TakenBy
Treats	Drug	Disease	{1, M}	HasIndication

3.5.6 The Instance Attribute Table

Instance attributes are those attributes that describe the instances of the concept and whose value(s) may be different for each instance of the concept. For each instance attribute, the ontologies must specify the following fields: its name; the concept it belongs to (attributes are local to concepts); its value type; and range of values (in the case of numerical values); Minimum and maximum cardinality.

3.5.6.1 *Patient Instance Attribute Table*

Table 3.22 shows the instance attribute table which consists of the instance attribute, for example, the patientID, address, DOB, Age, Gender, phone, FirstName, LastName, blood pressure, PulseRate, Weight, Temperature, Height, BodyMassIndex. The Concept name example is patient, the value type can be in integer decimal and string, the value range contains the value that is being assigned to the instance attribute, and the cardinality is a built-in OWL property that links a restriction class to a data value. This will be applied to another instance attribute such as the prescription instance attribute, Drug, Clinic, Pharmacy, Disease, Doctor, Receptionist, Pharmacist, Food and Allergy instance attribute.

Table 3-13 Patient Instance Attribute Table

Instance Attribute	Concept name	Value type	Value Range	Cardinality
PatientID	Patient	String	0-9	(1, 1)
Address	Patient	String	1-20	(1,1)
DOB	Patient	String	0-9	(1, 1)
Age	Patient	String	0-4	(1, 1)
Gender	Patient	String	0-9	(1,1)
Phone	Patient	String	0-8	(1, M)
FirstName	Patient	String	0-8	(1, 1)
LastName	Patient	String	0-9	(1, 1)
Blood Pressure	Patient	String	0-9	(1, 1)
PulseRate	Patient	Integer	0-2	(1, 1)

Weight	Patient	String	0-2	(1, 1)
Temperature	Patient	decimal	0-2	(1, 1)
Height	Patient	String	[0 3]	(1, 1)
BodyMassIndex	Patient	String	(0-9)	(1, 1)

3.5.6.2 Prescription Instances Attribute Table

Table 3.13 below shows the prescription instance attribute which contains the instance attribute example are Quantity, NextFill, DosageForm, DosageRoute, DosageStrength, Date, PrescriptionID, DispensedStatus, DDIStatus, FoodInteractionStatus, DiseaseInteractionStatus WrongDrugStatus InstructionQuantity instruction frequency InstructionDescription Price, the concept name, Value type, value range and cardinality.

Table 3-14 Prescription Instance Attribute Table

Instance Attribute	Concept name	Value type	Value Range	Cardinality
Quantity	Prescription	integer	1-120	(1,1)
NextFill	Prescription	integer	1-90	(1, 1)
DosageForm	Prescription	string	1-20	(1, 1)
DosageRoute	Prescription	string	1-20	(1,1)
DosageStrength	Prescription	string	1-100	(1, 1)
Date	Prescription	integer	1872-2022	(1, 1)
PrescriptionID	Prescription	string	0-9	(1, 1)
DispensedStatus	Prescription	string	1-20	(1, 1)
DDIStatus	Prescription	string	1-500	(1, 1)
FoodInteractionStatus	Prescription	string	1-20	(1, 1)
DiseaseInteractionStatus	Prescription	string	1-20	(1, 1)
WrongDrugStatus	Prescription	string	1-20	(1, 1)

InstructionQuantity	Prescription	integer	1-50	(1, 1)
InstructionFrequency	Prescription	integer	1-20	(1, 1)
InstructionDescription	Prescription	string	0-9	(1, 1)
Price	Prescription	string	0-9	(1, 1)

3.5.6.3 Drug instance Attribute Table

Table 3.14 shows the drug instance attribute table which consists of the drug instance attribute, examples are, the description, toxicity, SnomedCode, SideEffect, RouteOfElimination, ProteinBinding, Metabolism, MechanismOfAction, HalfLife, DrugName, Absorption, DosageRoute, DosageStrength, Alternative, Averagemass, Brand, UnitPrice, DosageForm, DrugBankI, the concept, Value type, Value Range(this is the set of values that the function assumes) and the cardinality(A restriction containing an owl cardinality constraint which describes a class of all individual that have exact N semantically distinct values for the property concerned where N is the value of the cardinality constraint).

Table 3-15 Drug Instance Attribute Table

Drugs Instance Attribute	Concept name	Value type	Value Range	Cardinality
Description	Drug	String	1-500	(1, 1)
Toxicity	Drug	String	1-500	(1, 1)
SnomedCode	Drug	String	1-20	(1, 1)
SideEffect	Drug	String	1-500	(1, M)
RouteOfElimination	Drug	String	1-500	(1, M)
ProteinBinding	Drug	String	1-500	(1, 1)
Metabolism	Drug	String	1-500	(1, M)
MechanismOfAction	Drug	String	1-500	(1, 1)
HalfLife	Drug	String	1-20	(1, M)
DrugName	Drug	String	1-500	(1, 1)
Absorption	Drug	String	1-500	(1, 1)
DosageRoute	Drug	String	0-9	(1, M)

DosageStrength	Drug	String	1-500	(1, M)
Alternative	Drug	String	1-20	(1, M)
AverageMass	Drug	String	1-500	(1, 1)
Brand	Drug	String	1-20	(1, M)
UnitPrice	Drug	String	0-9	(1, 1)
DosageForm	Drug	String	0-9	(1, M)
DrugBankID	Drug	String	0-9	(1, 1)

3.5.6.4 Clinic/Pharmacy Instance Attribute Table

Table 3.16 below shows the Clinic\Pharmacy instance attribute table which consists of the instance Attribute (Name, Address, Phone), Concept name (Clinic\Pharmacy), Value type (which can be string, integer, decimal etc.), Value range and the cardinality.

Table 3-16 Clinic / Pharmacy Instance Attribute Table

Instance Attribute	Concept name	Value type		Value Range	Cardinality
Name	Clinic/ Pharmacy	String		1-20	(1, 1)
Address	Clinic/ Pharmacy	String		1-20	(1, 1)
Phone	Clinic/ Pharmacy	String		0-9	(1, 1)

3.5.6.5 Disease Instance Attribute Table

Table 3.16 below shows the disease instance attribute table which consists of the Instance attribute (Name, Symptom), Concept Name which includes the disease instance attribute, the value type, value Range and cardinality.

Table 3-17 Disease Instance Attribute Table

Instance Attribute	Concept name	Value type	Value Range	Cardinality
Name	Disease	String	1-20	(1, 1)
Symptom	Disease	String	1-20	(1, M)

3.5.6.6 Doctor/ Pharmacist/ Receptionist

Table 3.17 shows the instance attribute of the Doctor\Pharmacist\Receptionist which consists of the instance attribute (FirstName, LastName), the concept Name, the value type, the Value Range and the cardinality.

Table 3-18 Instance Attribute of Doctor/Pharmacy/Receptionist

Instance Attribute	Concept name	Value type	Value Range	Cardinality
FirstName	Doctor Pharmacist Receptionist	String	1-9	(1, 1)
LastName	Doctor Pharmacist Receptionist	String	1-9	(1, 1)

3.5.6.7 Food/Allergy instance Attribute Table

Table 3.18 below shows the food/Allergy instance table which includes the instance attribute (Name), Concept Name, Value type, value Range and cardinality.

Table 3-19 Instance Attribute of Food/Allergy

Instance Attribute	Concept name	Value type	Value Range	Cardinality
Name	Food Allergy	String	0-9	(1, 1)

3.5.7 Class Attribute Table

The class attribute describes all the attributes in the class that is already included in the concept dictionary table each row simply describes the attribute of Table 3.19. The following information which should be included in the table is name, the concept that describes the name; value type; values(S); cardinality, and the instance attribute whose value attribute.

Table 3-20 Class Attribute Table

Class attribute names	Defined Concept	Value type	Cardinality
DDIStatus	Prescription	DDIExists, NoDDI	(1,2)
DispensedStatus	Prescription	NotDispensed, Dispensed	(1, 2)
DiseaseInteractionStatus	Prescription	NoDiseaseInteraction DiseaseInteractionExist	(1, 2)
FoodInteractionStatus	Prescription	NoFoodInteraction, FoodInteractionExist	(1,2)
DosageRoute	Prescription	Oral, Injection	(1, 2)
DosageForm	Prescription	Tablet, Capsule	(1, 2)
WrongDrugStatus	Prescription	CorrectDrugPrescribed, WrongDrugPrescribed	(1, 2)
Type	Appointment	Online, Inperson, Phone	(1, 3)
Gender	Person	Male, Female	(1,2)

3.5.8 Formula Table

Table 3.20 below shows the formula table of the IEPO ontology which includes the formula Name e.g. (Body Mass index, prescription Quantity, Prescription Price), Inferred Attribute, Formula and Description.

Table 3-21 Formula Table of the IEPO Ontology

Formula Name	Inferred Attribute	Formula	Description
Body Mass Index	Weight Height	$\text{BodyMassIndex} = \frac{\text{Weight}}{(\text{Height} * \text{Height})}$	To calculate the Body Mass Index of a Patient
Prescription Quantity	Instruction Quantity InstructionFrequency NextFill	Quantity = $\text{InstructionQuantity} * \text{InstructionFrequency} * \text{NextFill}$	To calculate the total number of tablets or unit drugs in a prescription
Prescription Price	UnitPrice Quantity	$\text{Price} = \text{UnitPrice} * \text{Quantity}$	The total cost of the prescription

3.5.9 Constant Table

For the proposed ontology, the Constant table of the healthcare interoperability e-prescription ontology is skipped because the proposed framework does not contain a Constant table.

3.5.10 Formal Axioms Table of the Clinic and Pharmacy Ontology

The proposed Methontology will describe the formal Axioms and rules in first-order logic. Table 3.22 shows the formal axioms in the e-prescription healthcare system that states that “A patient cannot be a doctor and a doctor cannot be a patient which shows incomparability between patient and doctor which corresponds to the referred concepts and the referred relations which are used inside the formal axiom. The variable used is for A patient and B for the doctor.

The formal axioms and logical expression may likely be difficult for an expert without the knowledge of the first-order logic. A logical-axioms table defines the concepts through logical expressions that are always true. Each defined axiom includes its name, its natural-language description, the concept to which the axiom refers, the attributes used in the axiom, the logical expression that formally describes the axiom using FOPC (first-order predicate calculus), and references.

Table 3-22 Logical-Axiom Table

Axioms name	Description	Expression	Referred concepts	Referred relations	Variable
Incomparability between Patient and Doctor	A patient cannot be a doctor and a doctor cannot be a patient.	not (exist(?A,?B) (Patient(?A) and Doctor(?B) [BelongsTo](?B,?A)))	Patient Doctor	belongs to	?A ?B
Incompatibility between Patient and Prescription	A prescription cannot be prescribed by a patient	not (exists(?A,?B) (Patient(?A) and Prescription(?B) [Prescribes](?A,?B)))	Prescription Patient	Prescribes	?A ?B
Drug to Drug	A single prescription	not (exists(?A,?B,?C) (Prescription(?A) and	Prescription	HasDrug	?A ?B

Interaction Incompatibility in a prescription	cannot have drugs that are interacting with each other	Drug(?B1) and Drug(?B2) HasDrug(?A, ?B1) and HasDrug(?A, ?B2) HasDrugInteraction(?B1, ?B2) → DDIStatus(?A, DDIExists)	Drug	HasDrugInteraction	?C
Drug to Drug interaction incomparability in two different prescriptions belonging to one patient	Two different prescriptions belonging to the same patient cannot have drugs that are interacting with each other's	Patient(P) ^ Prescription(?A) ^ Prescription(?A2) ^ Drug(?B1) ^ Drug(?B2) ^ BelongTo(?A1,?P) ^ BelongTo(?A2,?P) ^ HasDrug(?A1,?B1) ^ HasDrug(?A2,?B2) ^ HasDrugInteraction(?B1, ?B2)→ DDIStatus(?A, DDIExists)	Prescription Drug	HasDrug HasDrugInteraction	?A ?B ?C ?D
Incompatibility between Prescription and Receptionist	A prescription cannot be prescribed by a receptionist.	not (exists(?A,?B) (Receptionist(?A) and (Prescription(?B) [Prescribe](?A,?B)))	Prescription Receptionist	Prescribes	?A ?B
Incompatibility between Prescription and Patient	A prescription belongs to only one Patient	not(exist(?A,?B) (Prescription(?A) and PatientID(?B) [Belongsto](?A, ?B)))	Prescription Patient	Belongsto	?A ?B
Incompatibility between Patient, Disease and Drug.	A patient cannot be prescribed a drug for the wrong disease.	not(exist(?A, ?B,?C) (Patient(?A,) and (Drug(?B)and (Disease(?C) [Prescribed](?A, ?B))) [HasDrugFor](?A, ?C)))	Drug Disease Patient	Prescribed	?A ?B ?C
Incompatibility between Prescription, Drug and Dosage.	A Prescription cannot have a drug with the wrong amount of dosage	not(exist(?A,?B,?C) (Prescription(?A) and (Drug(?B) and (Dosage(?C) [HasDrug](?A, ?D))) [HasDrug](?A, ?B)))	Prescription Drug	HasDrug	?A ?B ?C

3.5.11 An Instance Table

An instance table lists the name, the attributes with known values in an instance, and the values of the above attributes, for each instance in the concept dictionary.

3.5.11.1 Concept Name: Patient

The individual or instance attribute for the patient class which includes the attribute address date of birth, age, gender, phone, firstname, lastname, blood pressure, pulserate, weight, temperature, height, bodymassindex, patientId with the name of the different instances is shown in Figure 3.22.

Table 3-23 Instance Attribute for the Patient Class

		Instance Name			
		JedaJoel	AbrahamJames	DorcasHope	AbelJohn
Attributes	Address	Plot 1600, Ext3, Palapye, Botswana	Plot1700, Phase 6, Gaborone, Botswana	Plot 2246, Ext10, Palapye, Botswana	Plot 2246, Area A, Francistown, Botswana
	DOB	1994-06-17T00:00:00	2000-07-18T00:00:00	1993-08-19T00:00:00	1980-08-20T00:00:00
	Age	28	22	29	42
	Gender	Male	Male	Female	Female
	Phone	71234565	72345612	75545655	76000156
	FirstName	Jeda	Abraham	Dorcas	Abel
	LastName	Joel	James	Hope	John
	BloodPressure	90/60mmHg	90/60mmHg	90/60mmHg	90/60mmHg
	PulseRate	100bpm	120bpm	80bpm	110bpm
	Weight	70	80	65	68
	Temperature	36,5°C	37,7°C	35,6°C	35,8°C
	Height	1.8	1.7	1.6	1.8
	BodyMassIndex	20.90	27.68	24.46	22.20
	PatientID	PA00000001	PA00000002	PA00000003	PA00000004

3.5.11.2 Instance Attribute for the Prescription Class

Table 3.23 below shows the individual of the concept prescription with the different PrescriptionID of the individual patient which includes the Quantity, NextFill, DosageForm, DosageRoute, DosageStrength, Date, PrescriptionID, DispensedStatus, DDISStatus, FoodInteractionStatus, DiseaseInteractionStatus, WrongDrugStatus, InstructionFrequency, InstructionDescription and Price and also the different patient descriptions.

Table 3-24 Concept Name

Individual	JedaJoelPR0000001	JedaJoelPR0000002	AbrahamJamesPR00000003		DorcasHopePR00000004	AbelJohnPR00000005
Quantity	90	60	90	30	90	60
NextFill	30	30	30	30	30	30
DosageForm	Tablet	Capsule	Tablet	Tablet	Capsule	Tablet
DosageRoute	Oral	Oral	Oral	Oral	Oral	Oral
DosageStrength	5	10	5	10	5	50
Date	2022-03-01	2022-04-01	2022-05-06	2022-05-06	2022-05-02	2022-06-02
PrescriptionID	PR00000001	PR00000002	PR00000003	PR00000003	PR00000004	PR00000005
DispensedStatus	NotDispensed	NotDispensed	NotDispensed	NotDispensed	NotDispensed	Dispensed
DDIStatus	DDIExists	DDIExists	DDIExists	DDIExists	NoDDI	NoDDI
FoodInteractionStatus	NoFoodInteraction	NoFoodInteraction	NoFoodInteraction	NoFoodInteraction	NoFoodInteraction	FoodInteractionExists
DiseaseInteractionStatus	NoDiseaseInteraction	NoDiseaseInteraction	NoDiseaseInteraction	NoDiseaseInteraction	NoDiseaseInteraction	DiseaseInteractionExists
WrongDrugStatus	CorrectDrugPrescribed	CorrectDrugPrescribed	CorrectDrugPrescribed	CorrectDrugPrescribed	WrongDrugPrescribed	CorrectDrugPrescribed
InstructionQuantity	1	1	1	1	1	1
InstructionFrequency	3	2	3	1	3	2
InstructionDescription	1 tablet, Morning, Noon, Night	1 tablet, Morning, Night	1 tablet, Morning, Noon, Night	1 tablet, Morning	1 tablet, Morning, Noon, Night	1 tablet, Noon, Night
Price	27	28,8	27	14,4	41,4	15,6

3.6 Intergration: Ontology Merging

Ontology merging is the process of joining two different ontologies together to expand and combine domain knowledge. Ontology merging can be done during the conceptualization phase or during the formalization phase. In this research, the merging will be done during the conceptualization phase. The clinic and pharmacy concept classification trees will be merged to form the IEPO classification tree which graphically represent the IEPO semi formal ontology

as shown in Figure 3-7. Merging is necessary enable querying of only one ontology which will accelerate the query time than when querying on several ontologies.

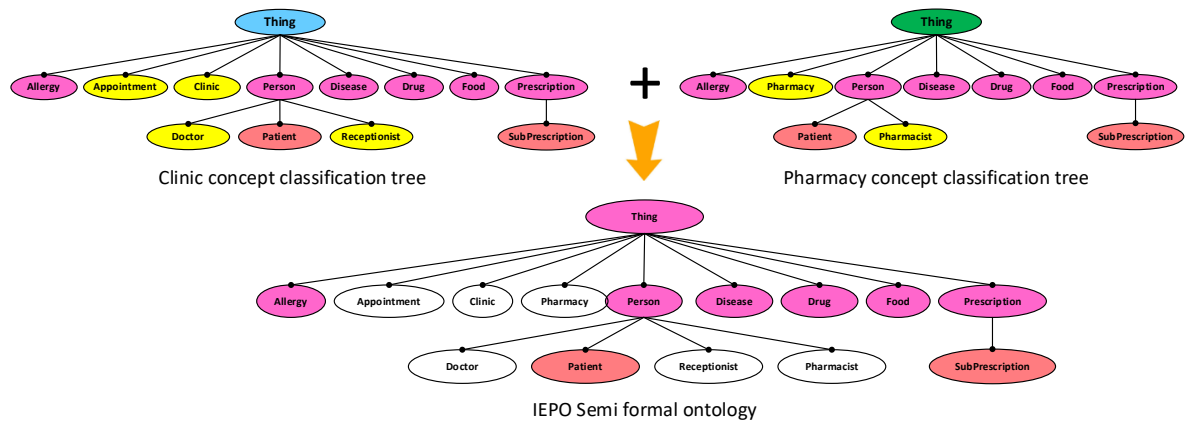


Figure 3-7: Merging of clinic and pharmacy concepts classification trees to form the IEPO semi formal ontology

3.7 Formalization

Figure 3-8 shows the IEPO formalization phase. Formalization is part of technical activities in Methontology presented in Section 3.1 on Figure 3-1.

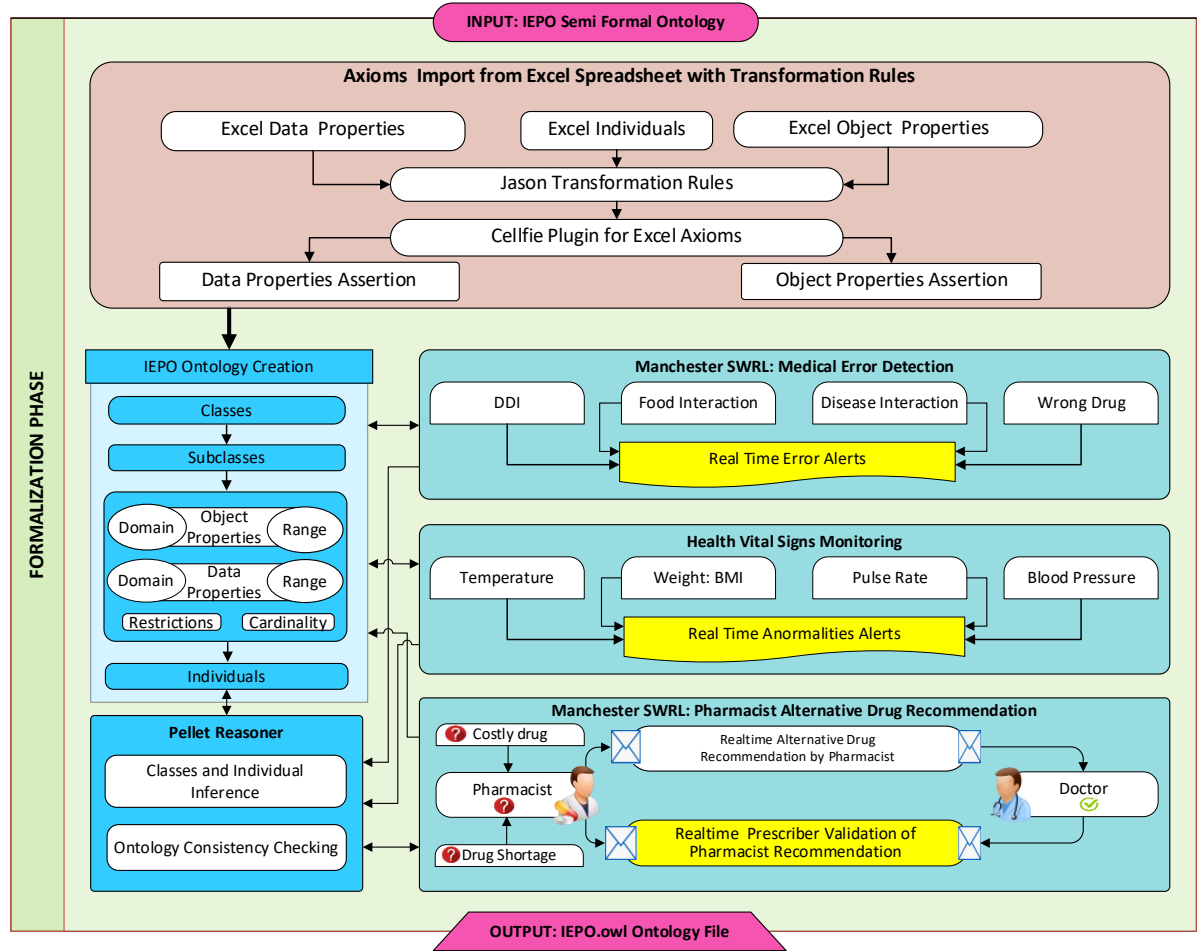


Figure 3-8. IEPO Formalization Phase

The formalization activity transforms the conceptual model presented from the previous section into a formal model using the chosen ontology development tools and languages. The output of formalization is an owl ontology file with an “.owl” file extension. The IEPO semi formal ontology will be converted to a formal ontology called “IEPO.owl”. The ontology language to be used is OWL2 and the justification is presented in Section 2.5.4 which is a summary of related works on ontology development languages. The ontology development tool adopted is Protégé which supports OWL2 and the justification is presented in the Literature Review Section 2.5.6 which presented a summary of related works on ontology development tools. Protégé is an open-source ontology editor and framework that is used to create intelligent systems at no cost.

The data from the conceptualization tables need to be manually created in the Protégé tool and this is a very tedious task which takes time. Therefore, the conceptualization tables will be created in an excel spreadsheet then batch imported into Protégé within a split of a second. The Cellfie plugin will be used to batch imported the data (axioms) from the spreadsheet to Protégé (OWL2). The classes, data properties, object properties and individuals are created manually in Protégé, they cannot be imported from spreadsheet using Cellfie. However, Linking classes and individuals with their data and object properties is so time consuming in Protégé when done manually and Cellfie plugin will be used to solved this issue by importin the subject-predicate-object axioms from excel for linking both classes (e.g. Patient HasPrescription Prescription) and individuals e.g., (Patient HasPrescription Prescription)

3.7.1 Class and subclass creation in Protégé

The IEPO ontology classes (See Section 3.2 for definition of terms) which were presented in Table 3-2 (Glossary of terms) like eg, Patient, are created in Protégé.

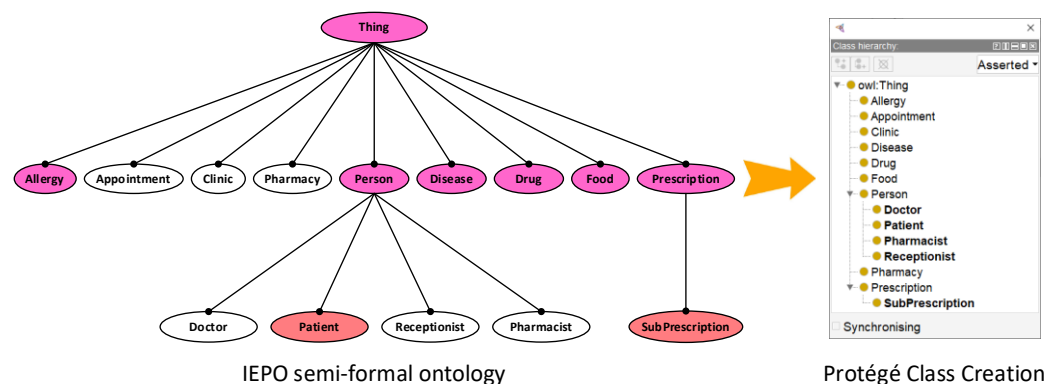


Figure 3-9 IEPO Class creation in Protégé: converting IEPO concept classification tree to Protégé class hierarchy.

To create the IEPO classes, the clinic concept classification tree in

Figure 3-3 and the pharmacy concept classification tree in Figure 3-4 were replicated in the Protégé Class Hierarchy window.

3.7.2 IEPO Object Properties Creation in Protégé

After creating the classes, there is a need to link them together to give the classes meaning on how they relate. The IEPO ontology object properties (See Section 3.2 for the definition of terms) presented in Table 3-3 will be created in the Protégé Object Property Hierarchy Window to link the IEPO ontology classes created in the previous Section 3.7.1 to form linked data, e.g., “HasPrescription.” Object properties will be used to link the individuals of the IEPO to be created later in this chapter. Figure 3-10 shows the creation of object properties in Protégé.

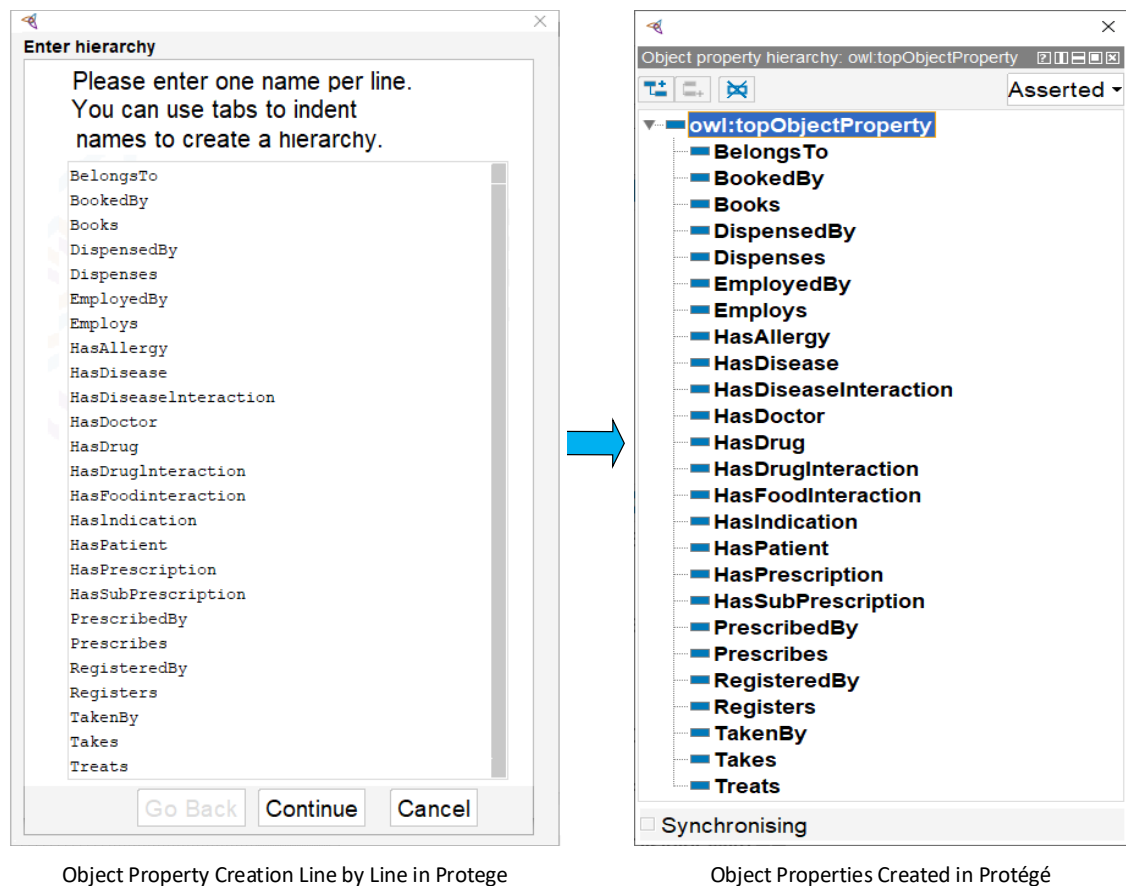


Figure 3-10: Creation of Object Properties in Protege

3.7.3 IEPO Data Properties Creation in Protégé

The instance attributes presented in the concept dictionary tables in Section 3.5.4 are called data properties in Protégé. Each class has several data properties, which were also presented in the instance attribute tables of each class in Section 3.5.6. The data properties were grouped

into their respective classes then each group was expanded, as shown in Figure 3-11, which shows the creation of data properties in Protégé.

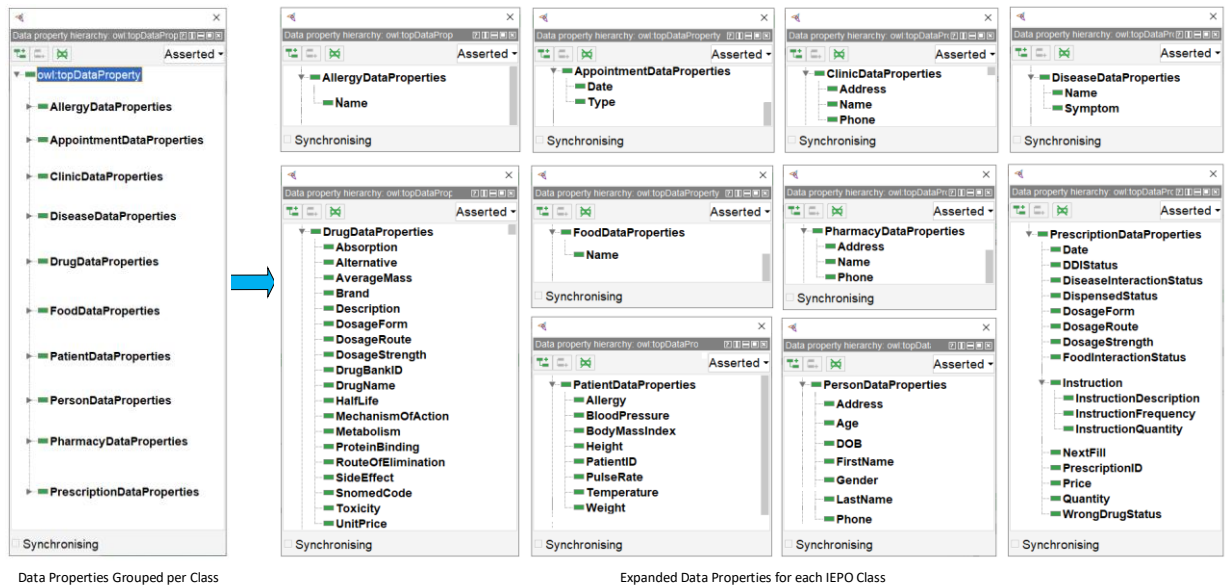
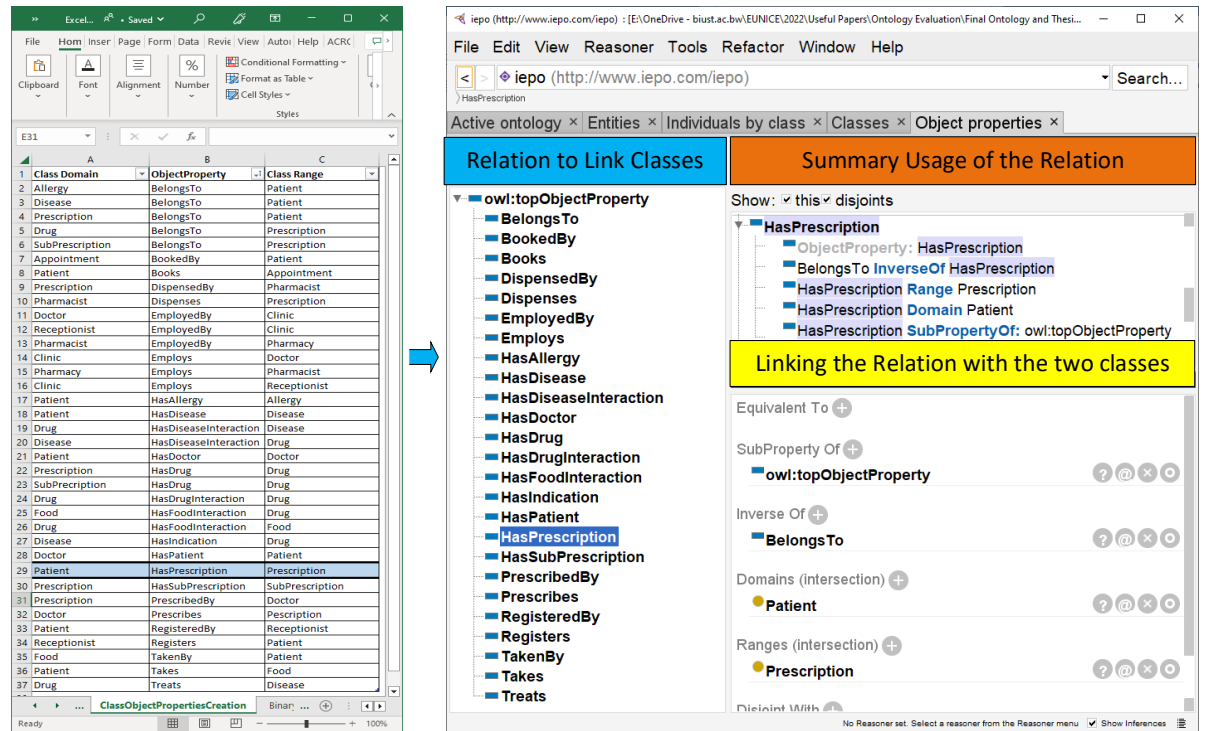


Figure 3-11. Creation of Data Properties in Protégé

3.7.4 Linking IEPO Classes Using Object Properties

For the IEPO classes to have meaning, they must be linked to form relationships using the object properties created, and Figure 3-12 shows how this was done in Protégé. There are many relations like “HasPrescription” linking the “Patient” class with the “Prescription” class such that the relationship becomes “Patient HasPrescription Prescription” meaning that a patient in IEPO has a prescription. An inverse of the relation “HasPrescription” becomes “BelongsTo” such that the inverse relation reads “Prescription BelongsTo Patient,” meaning that any particular prescription in IEPO always belongs to a particular patient. Figure 3-12 shows how the relation “HasPrescription” and its inverse “BelongsTo,” links the two classes “Patient” and “Prescription.” All the relations (object properties) between classes were done in excel and then similarly transferred to Protégé as shown in Figure 3-12.



Class Object Properties (Relations) Created First in Excel

Linking all relation from excel to their respective classes in Protege

Figure 3-12. Linking IEPO Classes Using Object Properties (Relations)

3.7.5 Creation of IEPO Individuals or Instances: Automatic Individual Import with Cellfie Plugin

This section transforms the conceptualization tables called “Instance tables” presented in Section 3.5.11 to Protégé individuals. The individuals are instances of the IEPO classes, e.g., the Patient class will have an individual or instance called Abel John. In Protégé, spaces are not allowed for individuals, so the individual becomes “AbelJohn.” All the individuals for each class to be created were presented in the conceptualization Section 3.5.11.

There are 39 IEPO individuals to be created in Protégé, which will be time-consuming due to the sequential nature of Protege. The individuals from Section 3.5.11. were all created in excel, which was faster and easier than in Protege. The Cellfie plugin a Protégé plugin, was used to batch import all the excel individuals within a few milliseconds, a task that would have taken longer. The Cellfie plugin batch imports the IEPO individual axioms using transformation rules, as shown in Figure 3-13. The transformation rules in the Cellfie plugin operate by accessing the

specified spreadsheet columns and then transforming the spreadsheet data on each row to Protégé axioms .

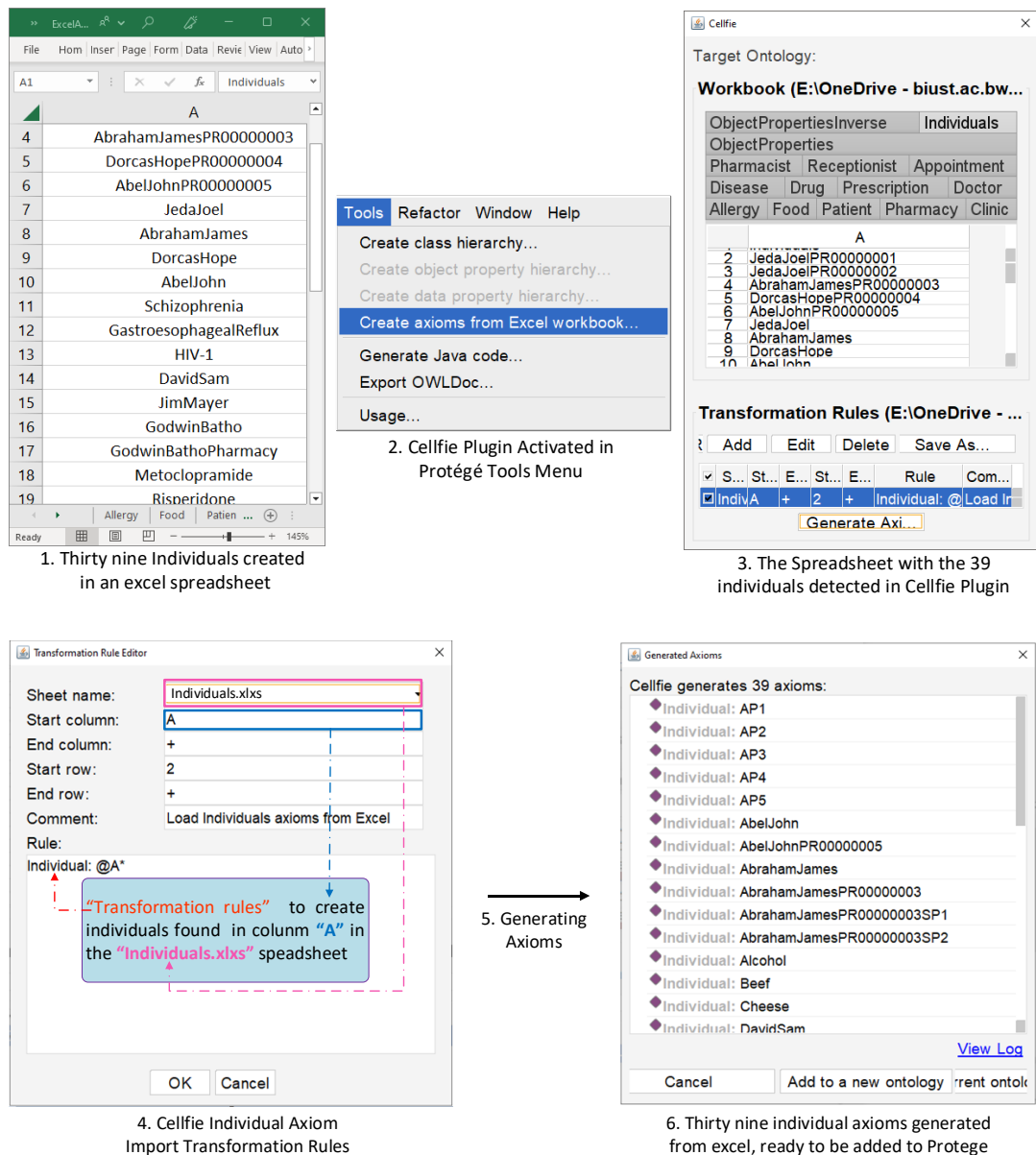


Figure 3-13. Creation of IEPO Individuals or Instances by importing the individual axioms from excel to Protégé using Cellfie plugin.

The transformation rule in Figure 3-13 specifies that the individuals will be extracted from only from column A of the “Individuals” spreadsheet. In addition, the starting row where the

individuals will be extracted are from row number 2 to infinity, that is anything in the column is added as an individual.

The generated individual axioms from excel are then appended to the existing IEPO ontology in Protégé and Figure 3-14 shows the imported individuals in Protégé grouped according to classes.

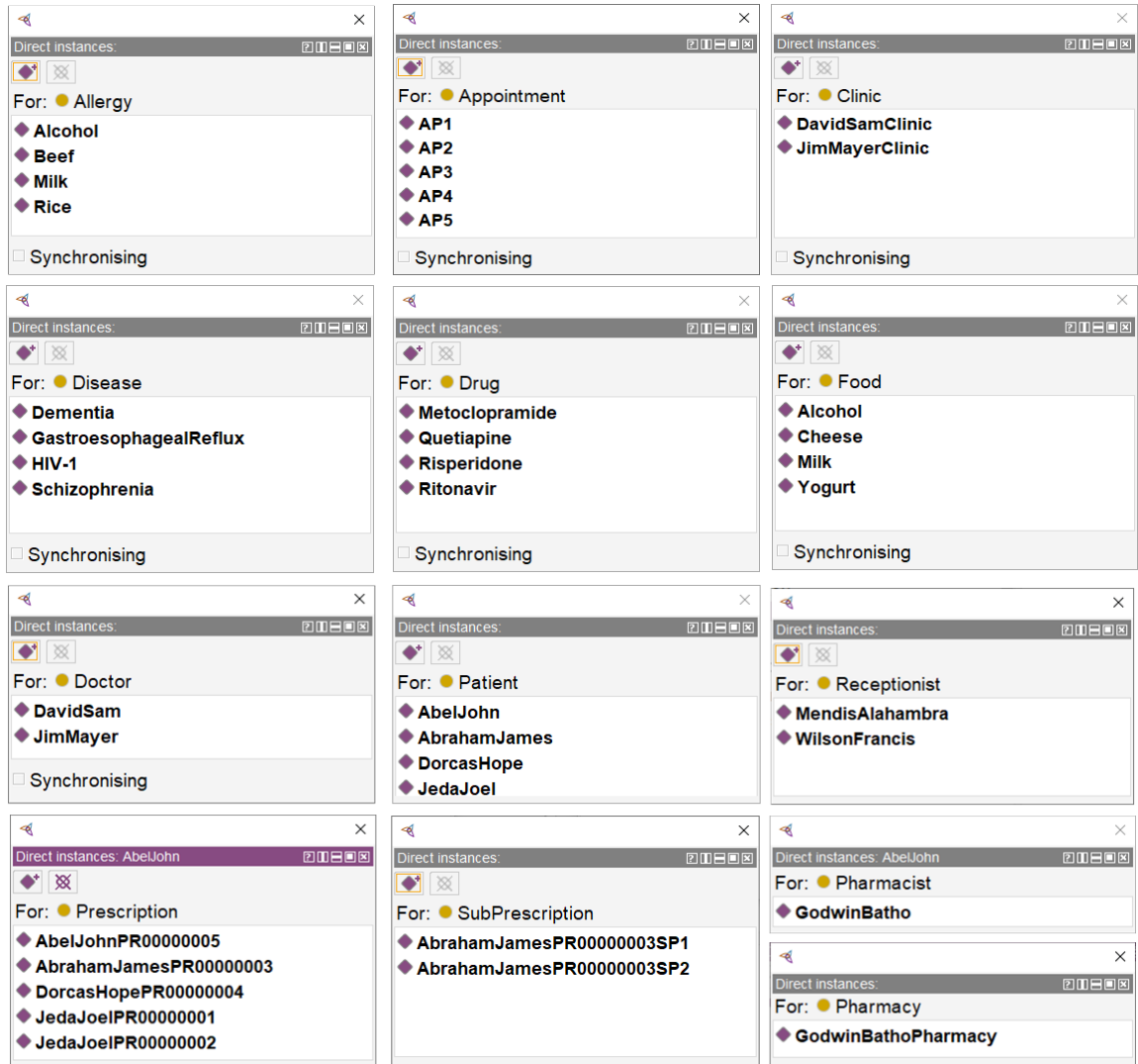


Figure 3-14. IEPO Individuals imported to Protégé using Cellfie plugin ad transformation rules.

3.7.6 Linking IEPO Individuals Using Object Properties via Cellfie Plugin

Just as the IEPO classes were linked using object properties, the individuals which are instances of this classes need to also be linked and associated with each other. This will give the

individuals meaning. The way the individuals associate with each other using object properties is similar to the association between classes. There are so many individuals and so many object properties in the EPO framework and it will be a tedious task to manually do linking in Protégé. The Cellfie plugin will be used to import the binary relation table from excel directly to Protégé. The individuals are linked using the domain and the range where the domain is the subject of a triple, and the range is the object while the object property is the predicate. The same procedure from Section 3.7.5 for creating individuals was followed for importing individuals' relations. Each object property has its own transformation rules which specifies the spreadsheets, the columns and the rows where data needs to be extracted from excel to Protégé. The transformation rules were saved as Jason files for reuse. Figure 3-15 shows how the IEPO individuals are linked with the IEPO object properties using Cellfie plugin.

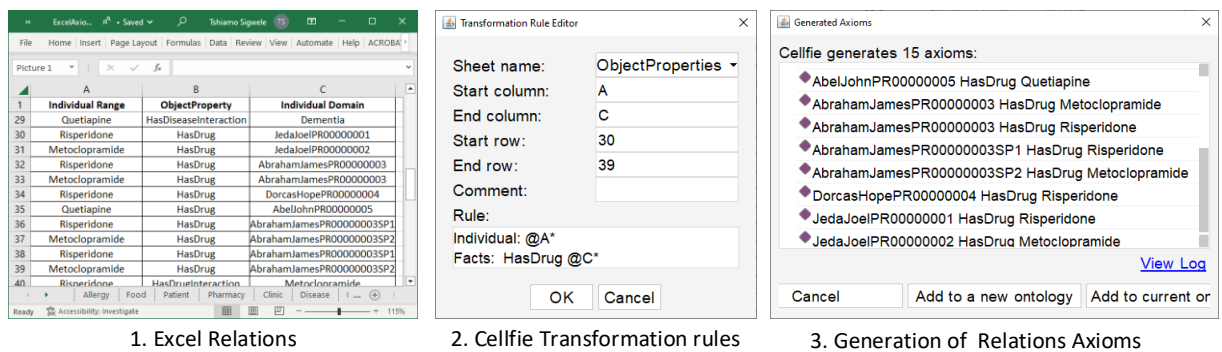


Figure 3-15. Importing and linking IEPO Individuals and relations into Protégé using Cellfie Plugin.

The transformation rule in figure X specifies that the individuals will be extracted from column A to column C of the object properties spreadsheet. In addition, the rows where the individuals will be extracted are from row 30 to row 39. "Facts" in the transformation rule indicates the object property which links the individuals from column A with the individuals from column C sequentially.

3.7.7 Creating Instances of Data Properties (Attributes) for Imported Individuals

All the individuals that were created in the previous section** have attributes or data properties. This individual data properties have instances which need to be created In the Protégé 2 and this instances of individual data properties we presented on instant tables in the

conceptualization phase section X. For example, a patient need to have a specific name age date of birth phone number etc. Since the conceptualization tables are captured in excel, this makes it easy to import into prodigy with the seal plugin. The instant tables will be imported into Protégé Following a similar approach as before for individuals' creation.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2	Individual	Address	DOB	Age	Gender	Phone	First Name	Last Name	Blood Pressure	Pulse Rate	Weight	Temperature	Height	Body Mass Index	Patient ID
3	JedaJoel	Plot 1600, Ext3, Palapye, Botswana	1994-06-17T00:00:00	28	Male	71234565	Jeda	Joel	90/60mmHg	100bpm	70	36,5°C	1.8	20.90	PA00000001
4	AbrahamJames	Plot1700, Phase 6, Gaborone, Botswana	2000-07-18T00:00:00	22	Male	72345612	Abraham	James	90/60mmHg	120bpm	80	37,7°C	1.7	27.68	PA00000002
5	DorcasHope	Plot 2246, Ext10, Palapye, Botswana	1993-08-19T00:00:00	29	Female	75545655	Dorcas	Hope	90/60mmHg	80bpm	65	35,6°C	1.6	24.46	PA00000003
6	AbelJohn	Plot 2246, Area A, Francistown, Botswana	1980-08-20T00:00:00	42	Female	76000156	Abel	John	90/60mmHg	110bpm	68	35,8°C	1.8	22.20	PA00000004

1. Instant Attributes Table for "Patient Class" from Conceptualization Phase created in Excel, to be imported to Protege

Transformation Rule Editor

Sheet name: Patient

Start column: A

End column: O

Start row: 3

End row: +

Comment: Import Patient Attributes Instances

Rule:

```

Facts: Address @B^(xsd:string)
Facts: DOB @C^(xsd:dateTime)
Facts: Age @D^(xsd:integer)
Facts: Gender @E^(xsd:string)
Facts: Phone @F^(xsd:string)
Facts: FirstName @G^(xsd:string)
Facts: LastName @H^(xsd:string)
Facts: BloodPressure @I^(xsd:string)
Facts: PulseRate @J^(xsd:string)
Facts: Weight @K^(xsd:decimal)
Facts: Temperature @L^(xsd:string)
Facts: Height @M^(xsd:decimal)
Facts: BodyMassIndex @N^(xsd:decimal)
Facts: PatientID @O^(xsd:string)

```

2. Transformation Rules to Import Patient's Excel Attributes Axioms to Protege



Property assertions: JedaJoel

Data property assertions +

- PatientID "PA00000001"^^xsd:string
- Weight 70
- Height 1.83
- FirstName "Jeda"^^xsd:string
- Gender "Male"^^xsd:string
- Address "Plot 1600, Ext3, Palapye, Botswana"^^xsd:string
- Phone "71234565"^^xsd:string
- Temperature "36,5°C"^^xsd:string
- LastName "Joel"^^xsd:string
- PulseRate "100bpm"^^xsd:string
- DOB "1994-06-17T00:00:00"^^xsd:dateTime
- BloodPressure "90/60mmHg"^^xsd:string
- BodyMassIndex 20.9
- Age 28

☐ Synchronising

3. An example of the Attributes for the Patient "Jeda Joel" created in Protege

Figure 3-16. 3.8.7 Creating Instances of Data Properties (Attributes) for Imported Individuals: An illustration of importing instances of data properties for the Patient called "Jeda Joel."

3.7.8 Prescription Medical Error Detection

As shown from Figure 3-8 , to detect prescription errors, rules called Semantic Web Rule Language (SWRL) are used in Protégé. Rules are a way of expressing additional things that OWL cannot express and inferred from the ontology, hence additional representation is required to manipulate the Protege OWL files with SWRL inference. When a DDI or a wrong drug is detected during prescribing, a real-time error alert is triggered. Table 3 in the 2 nd column shows the SWRL used to detect the errors. Below is the syntax of SWRL:

$$Male(?A) \wedge Male(?B) \wedge FatherOf(?A, ?B) \rightarrow SonOf(?B, ?A)$$

This statement reads, if there exists male person A and B and male person A is the father of male person B, then male person B is the son of male person A. With SWRL, the operator and inference connector are represented by the symbols $\wedge$ and $\rightarrow$ respectively. The left statements called atoms on the right side of the rightarrow are collectively called antecedent or body and the left are consequent or head. The intended meaning can be read as, whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold. The symbol "?A" symbol means a variable. DDI information is extracted from a 1.4GB XML DrugBank dataset [1] into IEPO. Drug Interaction Ontology (DINTO) was to be considered for reuse but it is not deprecated [33]. DDI error is addressed in two forms. First, as shown in Table 3, if single prescription A for any patient has two drugs B1 and B2 during prescribing, the SWRL will be triggered and the two drugs will be checked in the IEPO knowledge base for DDI. All drugs that interact are captured into IEPO from DrugBank and DRON. If the two drugs interact, e.g. $\text{HasDDI}(\text{?B1}, \text{?B2})$, then an error alert is triggered to the prescriber GUI that prescription A has drug interactions e.g. $\text{HasError}(\text{?A}, \text{DDIExits})$. Second, if a patient ? P has two different but current prescriptions ?A1 and ?A2 and ?A1 has a drug B1 and prescription ?A2 has a drug B2 and if there is DDI between drugs ?B1 and ?B2 eg. $\text{HasDDI}(\text{B1?}, \text{?B2})$, then an error alert is triggered and displayed to the prescribing doctor that prescription? A has a DDI error with the drugs involved specified. For a wrong drug error, a patient ? P with diagnosed sickness ?C1 cannot be prescribed a drug ? B which is used to treat a specific sickness ?C2 or else an error alert is displayed that prescription A has a wrong drug prescription.

3.7.9 Pharmacist Recommending Alternative Drugs

When a patient ? P goes to the pharmacy of her choice to collect medication which has been made When a pharmacist request in real-time during dispensing the validation of an alternative drug prescription from the doctor due to drug shortages or costly drugs to patent not covered by medical aid, and it is also very costly for the patient, the patient might opt to go to another pharmacy, but this may cause delays in case the patient's health is deteriorating and there is need for the medication to be taken for improved health.

SWRL is used as shown in Figure 3.11 and Table 3. First, at the pharmacy side, SWRL is used to check the availability of a drug B1 in prescription A belonging to patient P from the pharmacy inventory and its cost C1. If there is a need for the patient to get the medication instantly but the drug is not available and the nearest pharmacy is very far or if the drug is very costly for the patient and it is not covered by medical aid, the next action is triggered where the IEPO server is queried for an alternative drug B2 to drug B1 that cheaper using drug cost comparison rule `lessThan(?C2, ?C1)`. The pharmacist then sends a request via SMS, Whatsapp, email, or prescribing app, etc to the doctor for validation. In other words, the pharmacist is requesting the prescription drug B1 to be updated to the drug B2 and this can only be done by a prescriber/doctor. The pharmacy will then receive a notification that the request was approved and the new prescription drug can be dispensed. This drug recommendation process takes a matter of seconds without having to make a new prescription. Below is the SPARQL update query of the request which will be executed when the doctor approves by changing drug B1 represented as Risperidone to alternative and cheaper drug B2 as Quetiapine in the same prescription A. It should be noted that other properties for B1 dosage and instructions will be updated B2 properties A while the prescription identity is still A represented as Docus_Hope_001. The prefix called "link" is the SPARQL Endpoint which is an HTTP link pointing to the IEPO semantic server hosting all data.

3.7.10 Vital Signs Healthcare Monitoring

During prescribing, the patient's vital signs are monitored for abnormalities and anything below or above the threshold values triggers an alert signal to the prescriber who can advise the patient based on the alert. The following vital signs are monitored, temperature, body weight (BMI) pulse rate and blood pressure. The monitoring of vital signs is done using SPARQL queries and will be presented in the evaluation section under competence equations evaluation.

3.7.11 Prescription Statistical Analysis

The IEPO framework will perform statistical analysis for both clinical and pharmaceutical and patient data. The competency question present what needs to be analyzed hence they will

used to show what needs to be analyzed. SPARQL queries will be used to preform the analysis, for example, “How many prescription were prescribed by Doctor X in the year 2022?” The Statistical analysis via SPARQL queries will be presented in the evaluation section.

3.8 IEPO Implementation: IEPO Hosting and Prescription Information Dissemination

Figure 3-17 shows the framework IEPO implementation phase. The input to the implementation phase is the merged IEPO ontology (IEPO.owl file) which is the output of the formalization phase from the Protégé tool. For doctors, pharmacists, and patients to interoperate, there is a need to access and share the IEPO data seamlessly.

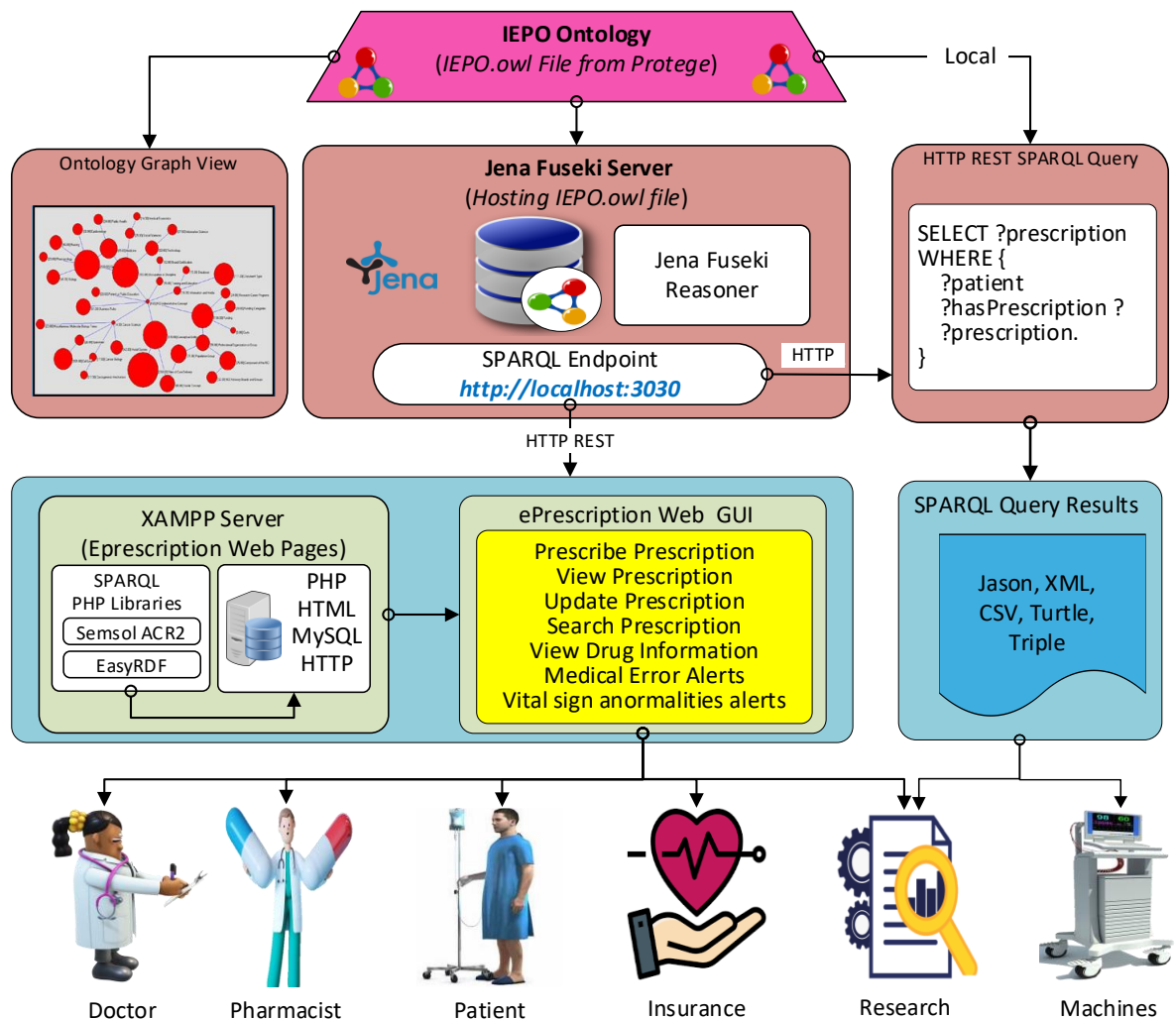


Figure 3-17. IEPO Implementation Framework

The IEPO prototype is a centralized prescribing web application where all healthcare entities can share healthcare information securely anywhere at any time and comprises of the following.

Ontology Graph View: The IEPO ontology file can be downloaded first then viewed using Webvowl tool at <http://vowl.visualdataweb.org/webvowl-old/webvowl-old.html#file=IEPO2.0.owl>

Jena Fuseki Server, SPARQL Query and SPARQL Endpoint: The IEPO ontology file is then hosted on an online Jena Fuseki server [35] as a Tripple Database (TDB) to be accessed seamlessly online via Hypertexts Transfer Protocol (HTTP) protocol. The data in the hosted IEPO file in the Jena Fuseki server is accessed via HTTP SPARQL Endpoint. A SPARQL Endpoint is a RESTful Application Programming Interface (API) for querying the IEPO ontology from the server using SPARQL. SPARQL is a recursive acronym for SPARQL Protocol and Resource Description Framework (RDF) Query Language. Representational State Transfer (REST) is used when querying the endpoint with SPARQL.

ePrescription Web GUI: Figure 3-18 shows the GUI of the proposed IEPO framework prototype that demonstrates interoperability with a secure login-based web application. An IEPO interoperability prototype was developed to demonstrate interoperability between healthcare systems entities. Doctors, pharmacists, patient etc. can perform the following via the GUI according to their roles; prescribe medication, update prescription, search prescription, view prescription, view drug information, view prescription medical error alerts and make amendments, view vital sign monitoring alerts. The prototype was hosted in a cross-platform, Apache, MySQL, PHP and Perl (XAMPP) server with a Graphical User Interface (GUI) to query the endpoint using SPARQL, Semsol ACR2 and EasyRDF PHP libraries. GUI users have roles like doctors, patients, pharmacists, etc., and the GUI renders pages according to the role. A patient has access to only their prescription information. Only doctors have a "Prescribe Medication" page.. The login uses a unique username and password which are stored on the MySQL server within XAMPP. The PHP application is hosted on a XAMPP server on Windows 10, 8GB PC. The GUI was accessed using iPad Pro connected to the same WiFi network as the IEPO XAMPP server with IP address 10.12.21.34 as shown in Figure 3-18, using a Chrome

browser. The server's IP address was identified by running the ipconfig command on the command console (CMD). Every entity or user can log in and access the specific information required via the internet. This web application is platform-independent and requires no installation as it can run in any web browser. Users can create, search, read, update, recommend, and delete prescriptions according to their roles. The GUI can be accessed, according to their roles, by pharmacists, doctors, patients, research groups, etc., to prescribe, view, update, search and view prescriptions and prevent prescription errors. Apart from the GUI, the hosted IEPO can also perform interoperability by querying via the SPARQL Endpoint remotely via the internet. The endpoint query results are in formats like JASON, XML, CSV, turtle, and triple, which can be sent to future virtual pharmacy dispensing machines and drones that can deliver to patients' homes without human intervention, hence saving time, cost and life.

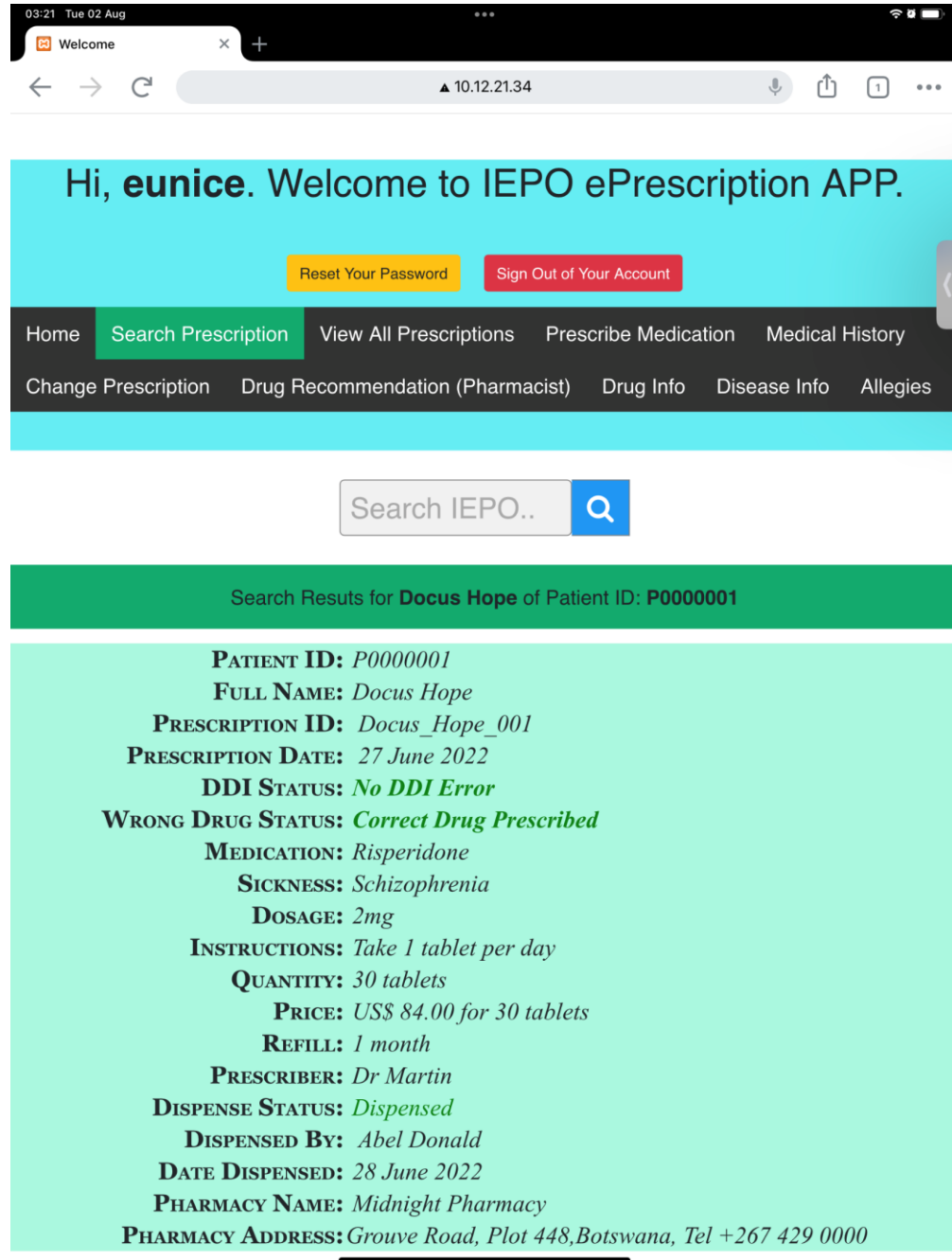


Figure 3-18. IEPO GUI.

3.8.1 Problems encountered during ontology development and their solutions.

1. Making changes to the ontology without frequently running the reasoner.

Solution: The pellet reasoner should be frequently run when there are changes so as not to compile error in your ontology which may lead to not being functional

2. Protégé kept freezing after running the Pellet reasoner when checking ontology inconsistencies, this was not caused by computer CPU or RAM shortage, nor by ontology size but by a bug in protégé 5.5.0 (The latest version at the time of the ontology development in 2022).

Solution: There should be a solution from the developer of the protégé editor

3. Manual creation of individuals that needs to be unique with their names auto-incrementing, e.g., Prescription1, then Prescription2, all the way up to Prescription1000 as the system prescriptions expand.

Solution: There is a need for automatic input when creating the individual to avoid manual inputs.

4. Making assertions of individual object properties was very tedious in protégé since if the instance of the range of the object property does not exist, then protégé will take us to the instances tab to create the individual, instead of just creating the individual under the individual object property assertion tab.

Solution: Simplicity when asserting the object properties whereby removing the range and making it directly to the instance tab to create individual than just creating under the individual object property assertion tab.

5. Protégé follows a class hierarchy such that when class A has subclass B, the instances of A are automatically inferred as instances of B. This is not true in real life since there can be a class called Prescription with subclass Drug, this does not mean that instances of class Drug are instances of Prescription.

Solution: The inference should be checked because another class or subclass cannot be merged to be of a different class.

6. Drug information was requested from drugbank.com, a 1.5GB XML file to try and import the XML file into protégé and automatically have drug with their drug

information but protégé could not accept XML files, even if a sample of the XML file was imported in case protégé was complaining because of the size of the XML file.

Solution: There was the need to convert the XML file to owl or RDF before reusing the XML tags and values into protégé, but this needed XML to RDF or OWL serialization which is a new area that needs further research.

7. Manual input of drug information from drugbank.com and drugs.com

Solution: Drug information from the drug bank should be automatically into the protégé editor.

8. Multiple click events to be able to manually populate a drug with drug information, that is, we had to move from one window after another to link an individual with its object and data properties.

Solution: This should be automatic to link an individual to their data or object properties

9. Manually linking individuals to object properties that can be easily inferred by a reasoner using SWRL to save time.

10. Merging of two different ontology duplicates itself instead of merging into the proposed ontology to be merged in protégé, so merging is not properly done.

CHAPTER 4 FINDINGS AND DISCUSSION: FRAMEWORK EVALUATION

Introduction

This chapter shows the results and discussion of the evaluation of the proposed framework. The proposed framework will adopt the following evaluation methods;

- The Ontology Domain Knowledge Coverage Evaluation: To ascertain the coverage of the e-prescription domain knowledge with their metrics e.g., precision, Hits, etc., the performance of the proposed framework was analyzed and compared to existing works.
- Medical Error Detection Evaluation: To detect medical errors like the DDI, wrong prescription, etc. for a particular disease using the SWRL.
- Vital Signs Healthcare Monitoring
- Ontology Evaluation using Ontology Pitfall Scanner: To check for the quality, correctness, and structure of the IEPO using the detailed custom metric with the categorized evaluation metric.
- IEPO Interoperability Metrics and prototype
- Ontology Competency Questions using SPARQL: To further test if the ontology works and was able to answer the following questions using the SPARQL.
- Other Ontology Metrics

There are many ways of evaluating an ontology framework. The first most common evaluation methods are domain knowledge coverage, ontology quality and correctness [67]–[71]. Ontology is also evaluated using competency questions [67], [72]–[74] to check if the ontology gives correct answers when queried using SPARQL. This research will evaluate the ontology e-prescription domain knowledge coverage, medical error detection accuracy, and also interoperability metrics.

4.0 Ontology Corpus Domain Knowledge Coverage Evaluation

The ontology Corpus domain knowledge is an open-source tool that is used in extracting information using the sketch engine based on e-prescription domain ontology.

The justification for using the corpus-based evaluation method is that it describes more words based on the description of word in searching for more related words based on the domain of knowledge or related topics [70]. The corpus-based evaluation approach in [70] is adopted for ontology evaluation in this work, Figure 2.4 shows the justification for its use. This approach was used to check if the IEPO covers the e-prescription domain in comparison to some related works. Also, the IEPO framework was evaluated for medical error detection performance as compared to the same related works. The IEPO will be compared with seven other e-prescription ontologies from the literature review. These seven ontologies are labeled as follows because most of them have no names: O1 in [75], O2 in [76], O3 in [77], O4 in [45], O5 in [78], O6 in [54] and O7 in [48].

Figure 4-1 shows the modified evaluation methodology adopted from [70] with each block explained in the subsequent sections. The evaluation from [70] which only considers a text corpus, was modified to also consider google images prescription samples described in Section 4.0.1. IEPO's correctness in terms of key consistency was validated using Protégé Pellet Reasoner without any errors.

4.0.1 Key Concept Extraction

An e-prescription corpus, which is a structured large text document containing all the relevant concepts about e-prescription from all possible prescriptions is created using the Sketch Engine platform [79]. Sketch Engine limits the corpus to 1 million words unless a fee is paid. However, 1 million words corpus is enough to demonstrate the evaluation of this research. Only important words or concepts which appear most in the corpus will be extracted as representing the domain of the corpus. Another disadvantage of Sketch Engine is that it only

considers text, not images, but prescription sample images are key to this research so a python script was written to automatically extract 500 prescription sample google images using image scrapping technology and GoogleImageScraper python package. The Optical Character Recognition (OCR) technology was then used to convert the prescription images to a text corpus. Word frequency was performed on both corpora and a list of words that appear most in both corpora was termed *RankedKeyConceptsList* list with normalized frequency weights in the range [0,1].

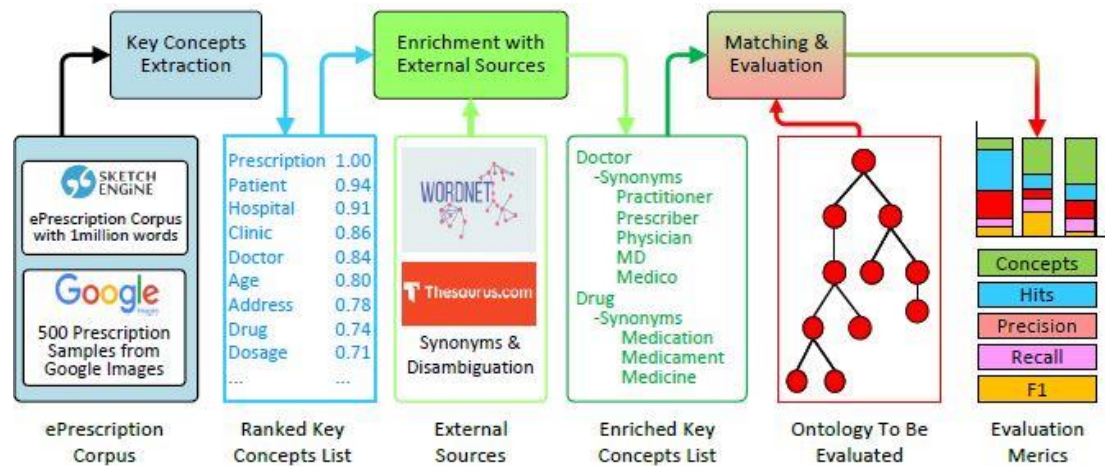


Figure 4-1 Modified Corpus-based Domain Knowledge Ontology Evaluation Methodology.

4.0.2 Enrichment with External Sources

The *RankedKeyConceptsList* list might contain words like physician or prescriber which are synonymous so, WordNet [80], an online platform is used to remove words disambiguity from the *RankedKeyConceptsList* list to form *EnrichedKeyConceptsList* list with synonyms. *EnrichedKeyConceptsList* now defines the whole knowledge domain of e-prescription.

4.0.3 Matching and Evaluation

This is where any ontology is evaluated by comparing its concepts called *ConceptsList* with *EnrichedKeyConceptsList*. If there are some common matching concepts found, the ontology then covers the e-prescription domain to a certain degree depending on further evaluation metrics discussed in Section 4.0.4.

4.0.4 Corpus-Based Evaluation Metrics

The IEPO ontology will be evaluated using the metrics *Concepts*, *Hits*, *Precision*, *Recall* and *F1*. These metrics have been wid.

ely used for ontology domain coverage evaluation, although they have been mainly employed for comparing a given ontology against a gold standard one [70].

4.0.4.1 Concepts Metric

Concepts Metric Concepts is the total number of ontology concepts. An ontology is defined by a finite set of concepts called *ConceptsList*, mainly a list of all ontology classes, subclasses, and their properties.

$$\text{ConceptsList} = \{\text{classes} \cup \text{subclasses} \cup \text{properties}\} \quad (1)$$

Concepts metric is the size or cardinality of *ConceptsList* which is the total number of ontology concepts. Concept is a positive integer. It is given as:

$$\text{Concepts} = |\text{ConceptsList}|, \text{Concepts} \in \mathbb{Z}^+ \quad (2)$$

The higher the value of *Concepts*, the higher the chances of an ontology covering more e-Prescription domain knowledge, and also the richer and more robust the ontology is.

4.0.4.2 Hits Metric

Hits is the number of concepts that are common in both the evaluated ontology concept list *ConceptsList* and *EnrichedKeyConceptsList* from the corpus. The higher the value of *Hits*, the higher the ontology covering more domain knowledge. The metric *Hits* is calculated as follows using a python script:

$$Hits = |ConceptsList \cap EnrichedKeyConceptsList| \quad (3)$$

4.0.4.3 Precision Metric

Precision is the ratio of *Hits* and $|ConceptsList|$. It is given as:

$$Precision = \frac{Hits}{|ConceptsList|}, Precision \in [0, 1] \quad (4)$$

4.0.4.4 Recall Metric

A *Recall* is the ratio of *Hits* and *EnrichedKeyConceptsList*;

$$Recall = \frac{Hits}{|EnrichedKeyConceptsList|}, Recall \in [0, 1] \quad (5)$$

4.0.4.5 F1 Metric

F1 metric normalizes Precision and Recall. It gives a more final performance of the ontology domain coverage. The higher the value of *F1*, the higher the ontology coverage of e-prescription domain knowledge.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, F1 \in [0, 1] \quad (6)$$

4.1 Result and Discussion for Corpus Evaluation

Table 4.1a shows the results after running a python script to calculate the performance metrics of IEPO and the 7 ontologies. Only 300 top concepts were extracted as *EnrichedKeyConceptsList*, these are concepts with a frequency factor greater than 0.01 when

using word frequency on the corpus, as stipulated in [34]. Table 4.1b shows the analysis of Table 4.1a results. It shows IEPO percentage performance compared to the seven ontologies. The green cells show that the proposed ontology performs better than the compared ontology by the percentage shown, and the red shows that the proposed ontology performs poorly.

As seen in Table 4.1b, for all the metrics, IEPO performs better than existing ontologies except for the -9% performance degradation of IEPO Precision metric compared with O3 which means that most concepts in O3 (11 Hits) are in the corpus, but Precision alone is misleading, because, an ontology with 11 concepts can perform better in terms of Precision if all the 11 concepts are in the corpus, but 11 concepts is a small number compared to 24 Hits in IEPO, hence the need for Recall and F1 metrics. IEPO performs better than O3 in terms of Recall by 50% due to more Hits in IEPO (24) compared to O3(11). The F1 metric is the final major metric to determine the performance of domain coverage because it finds the average of Precision and Recall. It can therefore be concluded that on average, using the F1 metric, IEPO is the best e-prescription ontology currently which performs 76% better than existing ePrescription ontologies in terms of ePrescription domain knowledge coverage.

Table 4.1a Corpus-Based Ontology Evaluation Results

Concepts	31	14	13	13	14	18	25	15
Hits	24	3	3	11	5	5	8	4
Precision	0.77	0.21	0.23	0.84	0.35	0.27	0.32	0.27
Recall	0.08	0.01	0.01	0.04	0.02	0.02	0.03	0.01
F1	0.14	0.02	0.02	0.07	0.03	0.03	0.05	0.02

Table 4-1b Domain Knowledge Coverage Performance

Metrics	O1	O2	O3	O4	O5	O6	O7	Mean
Concepts	55	58	58	55	42	19	52	48
Hits	88	88	54	79	79	67	83	77
Precision	73	70	-9	55	65	58	65	54

Recall	88	88	50	75	75	63	88	75
F1	86	86	50	79	79	64	86	76

4.2 Medical Error Detection Evaluation

The second evaluation is for medical prescription errors. The justification of the evaluation using medical error detection is to reduce the hospitalization of the Adverse Drug Effect which is the result of prescribing medical errors which reduces cost and some causality. There are 3 medical errors types that can be detected by IEPO: 1) **DDI Error 1** which is a DDI error detected within a single prescription belonging to a particular patient where 2 or more drugs are found to be interacting 2) **DDI Error 2** which is the DDI error detected in 2 or more prescriptions belonging to a single patient where 2 or more drugs within these prescriptions are interacting 3) **Wrong Drug Error 3** where a different drug is prescribed to treat a wrong sickness or condition. For each medical error type, 20 prescription errors were deliberately committed. Medical Error Detection Accuracy A will be used as a performance metric which is:

$$A = \frac{\text{No.of Type } M \text{ Medical Errors in Prescriptions}}{\text{No.of Type } M \text{ Medical Errors Detected}} \quad (7)$$

where M is either DDI Error 1, DDI Error 2, or Wrong Drug Error 3. A has values in the range $[0, 1]$ or percentage $[0, 100]$. The higher the value of A , the more accurate the detection of medical errors which means more safety for patients and reduced hospitalization from medical errors.

Table 4.2 shows the performance of the proposed IEPO framework in terms of prescription error detection accuracy. The results show that the framework was able to detect all there 3 prescription error types with 100% accuracy and give alerts to the prescriber. This will reduce the number of hospitalizations from ADEs caused by prescription medical errors, which in turn reduces medical costs and even some casualties. Other 7 e-prescription ontologies have no medical error detection as such they are not safe for adoption and reuse.

Table 4-2 Proposed IEPO Medical Prescription Error Detection Performance

Medical Errors	Error Metrics	IEPO	O1	O2	O3	O4	O5	O6	O7
DDI Error 1	No. Errors in Prescriptions	20	20	20	20	20	20	20	20
	No. Errors Detected	20	0	0	0	0	0	0	0
	Detection Accuracy, A	100%	0%	0%	0%	0%	0%	0%	0%
DDI Error 2	No. Errors in Prescriptions	20	20	20	20	20	20	20	20
	No. Errors Detected	20	0	0	0	0	0	0	0
	Detection Accuracy, A	100%	0%	0%	0%	0%	0%	0%	0%
Wrong Drug Error	No. Errors in Prescriptions	20	20	20	20	20	20	20	20
	No. Errors Detected	20	0	0	0	0	0	0	0
	Detection Accuracy, A	100%	0%	0%	0%	0%	0%	0%	0%

4.3 Vital Signs Healthcare Monitoring

The SPARQL queries on the SPARQL endpoint detect vital sign values which out of range and give alerts. Below are some the vital signs, typical values for the vital signs, value ranges of some abnormalities, and the SPARQL queries with results when executed on the SPARQL endpoint.

4.3.1 Body Mass Index: Overweight Alert

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX o: <http://www.iepo.com/iepo#>
SELECT ?Weight ?Height ?BMI ?BMISStatus WHERE {
  o:AbrahamJames o:Weight ?Weight.
  o:AbrahamJames o:Height ?Height.
  BIND ((?Weight/(?Height*?Height)) AS ?BMI).
  BIND (
    COALESCE(
      IF(?BMI > 24.9, "Overweight", 1/0),
      IF(?BMI >18.5 && ?BMI <= 24.9, "Normal Weight", 1/0),
      IF(?BMI<18.5, "Underweight", 1/0),
      1/0
    ) AS ?BMISStatus
  ).
}

```

SPARQL Results:

"80"^^<http://www.w3.org/2001/XMLSchema#decimal>"1.7"^^<http://www.w3.org/2001/XMLSchema#decimal>"27.681660899653979238754325"^^<http://www.w3.org/2001/XMLSchema#decimal>"Overweight"
--

Figure 4-2. BMI results showing that a patient with 80kg weight, 1.7m height has 27.68 BMI, which is overweight

4.3.2 Temperature Monitoring: Covid-19 Alert

The prefixes in the SPARQL's are similar for all the SPARQL queries so they will be eliminated from this point.

```
SELECT ?Age ?Temperature ?AgeStatus WHERE {
  o:AbrahamJames o:Age ?Age.
  o:AbrahamJames o:Temperature ?Temperature.
  BIND (
    COALESCE(
      IF((?Age > 0 && ?Age <=1) && ?Temperature > 38, "Your infant's temperature is very high, there is a high probability you have COVID-19, contact GP", 1/0),
      IF((?Age > 1 && ?Age <=12)&& ?Temperature > 39, "Your child's temperature is very high, there is a high probability you have COVID-19, contact GP", 1/0),
      IF((?Age > 12 && ?Age <=150)&& ?Temperature > 39.4, "Your temperature is very high, there is a high probability you have COVID-19, contact GP", 1/0),
      "You temperature is within the normal range."
    ) AS ?AgeStatus
  ).
}
```

SPARQL Results:

"1"^^<http://www.w3.org/2001/XMLSchema#integer>	"37.7"^^<http://www.w3.org/2001/XMLSchema#decimal>	"You temperature is within the normal range."
---	--	---

Figure 4-3. COVID-19 monitoring: a patient with the correct temperature.

4.3.3 Pulse Rate Monitoring Alerts (Blood pressure follows the same procedure)

```
SELECT ?Age ?PulseRate ?PulseRateStatus WHERE {
  o:AbrahamJames o:Age ?Age.
  o:AbrahamJames o:PulseRate ?PulseRate.
  BIND (
    COALESCE (
      IF((?Age > 0 && ?Age <=1) && (?PulseRate < 100 || ?PulseRate >180), "Your infant's PulseRate is abnormal, contact a GP", 1/0),
      IF((?Age > 1 && ?Age <=12) && (?PulseRate >=80 && ?PulseRate <=120), "Your child's PulseRate is abnormal, contact a GP", 1/0),
      IF((?Age > 12 && ?Age <=150) && (?PulseRate >=60 && ?PulseRate <=100), "Your PulseRate is abnormal, contact your GP", 1/0),
      "The PulseRate is within the normal range."
    ) AS ?PulseRateStatus
  ).
}
```

```

    ) AS ?PulseRateStatus
  ) .
}

```

Results:

Age	PulseRate	PulseRateStatus
"22"^^<http://www.w3.org/2001/XMLSchema#integer>	"120"^^<http://www.w3.org/2001/XMLSchema#integer>	"The PulseRate is within the normal range."

Figure 4-4 Results of a patient with a regular pulse rate within the normal range.

4.4 Prescription Analysis

Prescription analysis analyzes the prescription within the IEPO, there are many things to analyze but just two examples will be shown using SPARQL queries.

Analysis Question 1: Give a list of patients born between August 1993 and 20th August 1980.

Sparql Query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?subject ?DOB
  WHERE
{
  ?subject <http://www.iepo.com/iepo#DOB> ?DOB
  FILTER (?DOB < "1993-08-20T00:00:00"^^xsd:dateTime)
  FILTER (?DOB > "1980-08-20T00:00:00"^^xsd:dateTime)
}

```

Results:

subject	DOB
DorcasHope	"1993-08-19T00:00:00"^^<http://www.w3.org/2001/XMLSchema#dateTime>

Figure 4-5 Analytical results showing the patient Docus Hope born on 19th August 1993.

Analysis Question 2: Give a list of all patients whos age is greater than 30 years.

Sparql Query:
 SELECT ?subject ?Age

```

WHERE
{
  ?subject <http://www.iepo.com/iepo#Age> ?Age
FILTER (?Age >30)
}

```

Results:

subject	Age
AbelJohn	"42""^<http://www.w3.org/2001/XMLSchema#integer>

Figure 4-6 Figure 4-7 Analytical results showing the patient Abel John with 42 years.

4.5 USABILITY RATING

Figure 10 shows
the usability evaluation results of IEPO application.

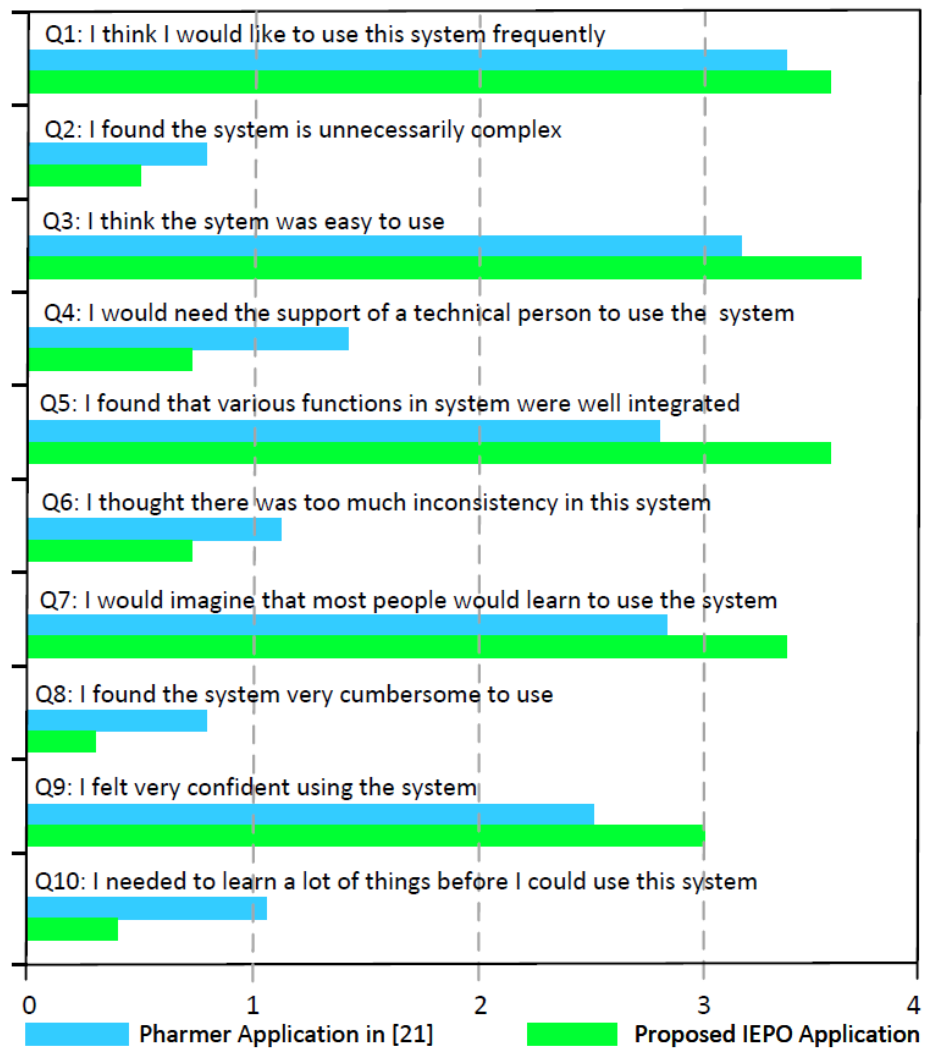


Figure 4-8 Usability Evaluation from for IEPO compared to Pharmer Application for O5 ontology. (0: Strongly disagree, 1: Disagree, 2: Neutral, 3: Agree, 4: Strongly agree). Odd-numbered Questions are worded positively and even-numbered Questions are worded negatively.

The usability metric determines how usable a software application is to the user. Participants were asked to interact with the IEPO web application and afterwards asked to fill out a questionnaire which consisted of two parts of 10 questions with questions shown in Figure 4-8: feature usage questions and usability experience questions using the System Usability Scale (SUS). The SUS system is used to grade the usability of IEPO. SUS is a standardized, simple, ten-item Likert scale-based questionnaire giving a global view of

subjective assessments of usability. For positively-worded items (1, 3, 5, 7 and 9), the score contribution is the scale position minus 1. For negatively-worded items (2, 4, 6, 8 and 10), it is

5 minus the scale position. To get the overall SUS score, multiply the sum of the item score contributions by 2.5. Thus, SUS scores range from 0 to 100 in 2.5-point increments and the greater the SUS score, the more usable the application is. The usability evaluation results for IEPO application were compared with Ontology 5 (O5) semantic application called Pharmer. The results showed a mean usability score of 75% for Pharmer and 87% for IEPO. This shows that IEPO application performs 12% better than Pharmer in terms of usability by users. In addition, 87% shows the system has a very good level of usability.

4.6 Ontology Evaluation using Ontology Pitfall Scanner

The Ontology Pitfall Scanner (OOPS) is an open-source and online based ontology evaluation tool developed by (Villalon: February 2016). The choice of using OOPS for the evaluation of this research can be justified by the result in Table 2-6. OOPS is used in detecting pitfalls within an ontology and giving suggestions on how to correct the pitfalls. OOPS has a total of 41 pitfalls but can currently detect only 33 pitfalls while the other 8 are inactive as they are being updated. The pitfalls are shown in the Appendix section. The pitfall is assigned levels of importance which include critical (red), important (orange) and minor (yellow). Below is a snapshot of the description of pitfall levels of importance.

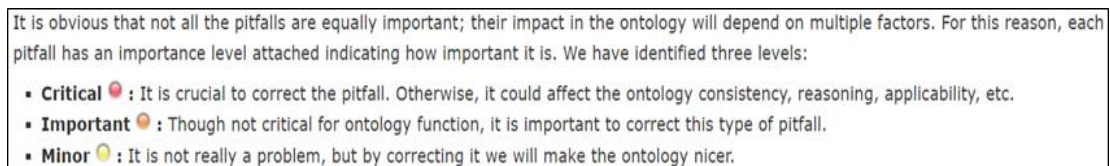


Figure 4-9 OOPS Pitfalls level of importance

Figure 4-10 shows the OOPS landing page interface. In the initial step of evaluation, a URI where the ontology is hosted is pasted under the “Scanner by URI” textbox, or the ontology is pasted into the “Scanner by direct input” text field.

OOPS! Ontology Pitfall Scanner!

OOPS! (Ontology Pitfall Scanner!) helps you to detect some of the most common pitfalls appearing when developing ontologies. To try it, enter a URI or paste an OWL document into the text field above. A list of pitfalls and the elements of your ontology where they appear will be displayed.

Scanner by URI:

Example: `http://oops.linkeddata.es/example/swc 2009-05-09.rdf`

Scanner by direct input:

If you just include the RDF code here, the following Pitfalls will not be checked:
P36. URI contains file extension, P37. Ontology not available, P40. Namespace hijacking

☐ Uncheck this checkbox if you don't want us to keep a copy of your ontology.

☐ Select Pitfalls for Evaluation ☐ Select Category for Evaluation

[Go to simple evaluation](#)

Figure 4-10 The OOPS Landing Page Interface.

In OOPS, an ontology can be evaluated in two ways: selecting pitfalls or categories. The category evaluation in Figure 4.4 was chosen as it is more detailed with relevant and commonly used ontology metrics which are classified into two: classification by dimension and classification by evaluation criteria which will be described in the following sections.

☐ Select Pitfalls for Evaluation ☒ Select Category for Evaluation

Classification by Dimension

☐ **Structural Dimension**

- ☐ **Modelling Decisions:** Checks for pitfalls P02, P03, P07, P21, P24, P25, P26 and P33.
- ☐ **Wrong Inference:** Checks for pitfalls P05, P06, P19, P27, P28, P29 and P31
- ☐ **No Inference:** Checks for pitfalls P11, P12, P13 and P30.
- ☐ **Ontology language:** Checks for pitfalls P34, P35 and P38.

☐ **Functional Dimension**

- ☐ **Real World Modelling or Common Sense:** Checks for pitfall P04 and P10.
- ☐ **Requirements Completeness:** Checks for pitfall P04 and P09.
- ☐ **Application context:** Checks for pitfalls P36, P37, P38, P39 and P40.

☐ **Usability-Profiling Dimension**

- ☐ **Ontology Clarity:** Checks for pitfalls P08 and P22.
- ☐ **Ontology Understanding:** Checks for pitfalls P02, P07, P08, P11, P12, P13, P20, P32 and P37
- ☐ **Ontology Metadata:** Checks for pitfalls P38 and P41

Classification by Evaluation Criteria

☐ **Consistency**

For this evaluation criteria the following pitfalls will be checked: P05, P06, P07, P19 and P24.

☐ **Completeness**

For this evaluation criteria the following pitfalls will be checked: P04, P10, P11, P12 and P13.

☐ **Conciseness**

For this evaluation criteria the following pitfalls will be checked: P02, P03 and P21.

Figure 4-11 Evaluation by Categories

The IEPO 2.0 owl file was opened with notepad, copied, and pasted onto OOPS. Each OOPS category was evaluated sequentially and corrected in the Protege. In the discussions below, the following terms will be used interchangeably to mean the same thing, relation or object property, concept or class, attribute or data property, and instance or individual.

4.6.1 Classification by Evaluation Criteria

The classification by evaluation criteria evaluates the ontology based on three metrics: consistency, completeness, and conciseness, described below.

4.6.2 Classification by Evaluation Criteria: Consistency Metric

Consistency checks the ontology for any contradictions. The pitfalls to check are P05, P06, P07, P19, and P24. If any pitfall is detected, then the ontology is not consistent to some degree, depending on the number of pitfalls detected. Figure 4.5 shows the results for the consistency metric. Pitfalls P06 and P07 successfully passed the evaluation test with no errors detected as shown in Figure 4.6. Inconsistencies were detected on pitfalls P05, P19 and P24.

Evaluation results

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- **Critical** 🚫 : It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- **Important** ⚠️ : Though not critical for ontology function, it is important to correct this type of pitfall.
- **Minor** 🟡 : It is not really a problem, but by correcting it we will make the ontology nicer.

[\[Expand All\]](#) | [\[Collapse All\]](#)

Results for P05: Defining wrong inverse relationships.	3 cases Critical 🚫
Results for P19: Defining multiple domains or ranges in properties.	11 cases Critical 🚫
Results for P24: Using recursive definitions.	1 case Important ⚠️

Figure 4-12 Result for consistency metric under classification by evaluation criteria

Congratulations!

Your ontology does not contain any bad practice detectable by OOPS! from the ones you have chosen.

Figure 4-13 Result for pitfalls P06 and P07

The results show that P05 and P19 are critical and need to be corrected while P24 has 1 ontology element using a recursive definition and it is also important to correct. P19 has 11 cases or elements where multiple domains, and ranges are defined. For P05, there are three cases where wrong inverse relationships were defined and P05 is a critical pitfall crucial to correct. Each pitfall can be expanded to show the ontology elements where the pitfalls occurred, which enables the developer to correct the pitfalls. For example, P05 results show that the relation `HasDiseaseIntercation` is made the inverse of `HasDrugInteraction` which is a

mistake because the relation HasDiseaseIntercation has no inverse relation as it is symmetric. This will be corrected by removing HasDiseaseIntercation as the inverse of HasDrugInteraction. In addition, the object properties HasDrugInteraction and HasFoodInteraction were made to be inverse relations in protégé but they should be symmetric relations because, if Risperidone HasDrugInteraction with Metoclopramide, then Metoclopramide HasDrugInteraction with Risperidone. This means that the symmetric property of relations overrides the inverse property and that the inverse property is already taken care of by the symmetric property. This pitfall will be corrected to change HasDiseaseInteraction, HasDrugInteraction and HasFoodInteraction from being inverse relations to being symmetric. The BelongsTo object property is detected as maybe not being the inverse of the HasDrug relation, this is because the axiom “SubPrescription HasDrug Drug” was created in protégé but its inverse which is “Drug BelongsTo SubPrescription” was not created. This will be corrected by adding SubPrescription class as the range of the relation BelongsTo in Protégé.

Results for P05: Defining wrong inverse relationships.	3 cases Critical 🚨
Two relationships are defined as inverse relations when they are not necessarily inverse.	
• This pitfall appears in the following elements: > http://www.iepo.com/iepo#HasDiseaseInteraction may not be inverse of http://www.iepo.com/iepo#HasDrugInteraction > http://www.iepo.com/iepo#BelongsTo may not be inverse of http://www.iepo.com/iepo#HasDrug > http://www.iepo.com/iepo#HasFoodInteraction may not be inverse of http://www.iepo.com/iepo#HasFoodInteraction	
Results for P19: Defining multiple domains or ranges in properties.	11 cases Critical 🚨
Results for P24: Using recursive definitions.	1 case Important ⚠️

Figure 4-14 Pitfall P05 result expanded to show the affected ontology elements

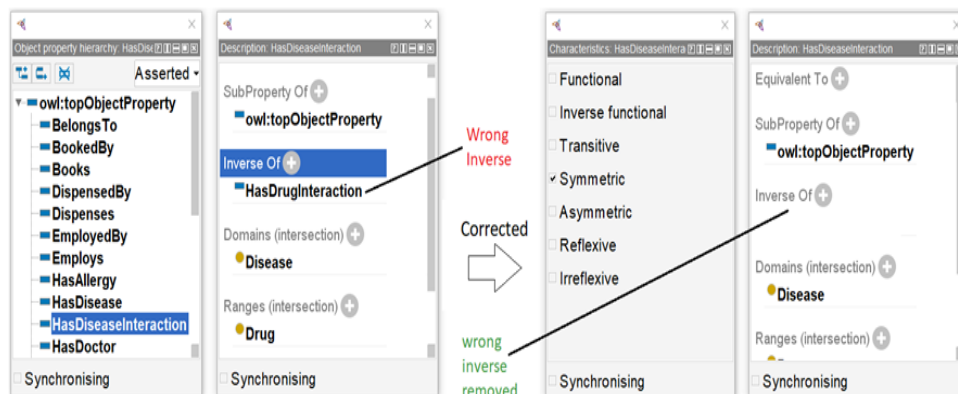


Figure 4-15 Correction of Pitfall in Protégé: Removing wrong inverse relationships

On expanding P19 as shown in **Figure 4-16**, the critical pitfall is that the 11 object properties have multiple domains and ranges which is true. For example, the object property *EmployedBy* has three domains (source concepts) which are *Doctor*, *Pharmacist* and *Receptionist* and two ranges (target concepts) which are *Clinic*, and *Pharmacy* as shown in the binary relation table below. OOPS complains that the ontology is not consistent as an object property should not have more than one domain or range. But in OWL, multiple domains and ranges for a single object property are allowed and this is called domain intersection or range intersection, as this critical pitfall will not be corrected as it is consistent giving with the owl language used with no inconsistencies even after running the Pellet reasoner in Protégé to check inconsistencies (Source: <http://protegeproject.github.io/protege/views/object-property-description/>).

Results for P19: Defining multiple domains or ranges in properties.
11 cases | Critical ❗

The domain or range (or both) of a property (relationships and attributes) is defined by stating more than one `rdfs:domain` or `rdfs:range` statements. In OWL multiple `rdfs:domain` or `rdfs:range` axioms are allowed, but they are interpreted as conjunction, being, therefore, equivalent to the construct `owl:intersectionOf`. This pitfall is related to the common error that appears when defining domains and ranges described in [7].

- This pitfall appears in the following elements:

> http://www.iepo.com/iepo#EmployedBy > http://www.iepo.com/iepo#HasDrug > http://www.iepo.com/iepo#BelongsTo > http://www.iepo.com/iepo#Employs > http://www.iepo.com/iepo#DosageForm > http://www.iepo.com/iepo#DosageStrength	> http://www.iepo.com/iepo#Phone > http://www.iepo.com/iepo#DosageRoute > http://www.iepo.com/iepo#Name > http://www.iepo.com/iepo#Address > http://www.iepo.com/iepo#Date
--	--

Figure 4-16 Pitfall P19 results expanded to show the effected ontology elements.

Object property hierarchy: BelongsTo

- owl:topObjectProperty
- BelongsTo
- BookedBy
- Books
- DispensedBy
- Dispenses
- EmployedBy
- Employs
- HasAllergy
- HasDisease
- HasDiseaseInteraction
- HasDoctor
- HasDrug
- HasDrugInteraction
- HasFoodInteraction
- HasIndication
- HasPatient
- HasPrescription
- HasSubPrescription
- PrescribedBy

☐ Synchronising

Description: BelongsTo

Inverse Of +

- HasPrescription
- HasDisease
- HasAllergy
- HasDrug
- HasSubPrescription

Domains (intersection) +

- Drug
- SubPrescription
- Prescription
- Allergy
- Disease

Ranges (intersection) +

- Patient
- Prescription
- SubPrescription

☐ Synchronising

No Correction required

➔

Description: BelongsTo

Inverse Of +

- HasPrescription
- HasDisease
- HasAllergy
- HasDrug
- HasSubPrescription

Domains (intersection) +

- Drug
- SubPrescription
- Prescription
- Allergy
- Disease

Ranges (intersection) +

- Patient
- Prescription
- SubPrescription

☐ Synchronising

Figure 4-17 The protégé Object property

In **Figure 4-16** no correction is required for pitfall P19 in protégé IEPO is an owl ontology and OWL accepts multiple domains and range properties.

Finally, under the consistency metric, when expanding P24, the results are shown in **Figure 4-18** and marked as “important”. The element or relation affected is *HasFoodInteraction* with the axiom “*Drug HasFoodInteraction Food.*” The pitfall is detected as recursion because the relation was made the inverse of itself. This error was corrected by removing its recursive inverse property and also making it symmetric as shown in Figure. Asymmetric property means a relation is an inverse of itself.

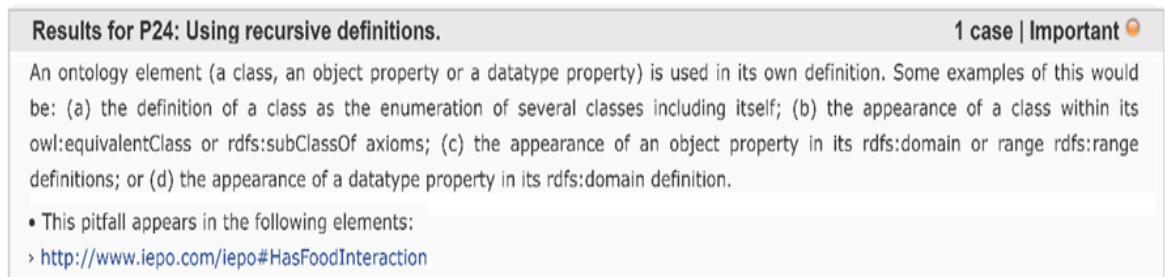


Figure 4-18 Pitfall P24 results expanded to show the affected ontology elements.

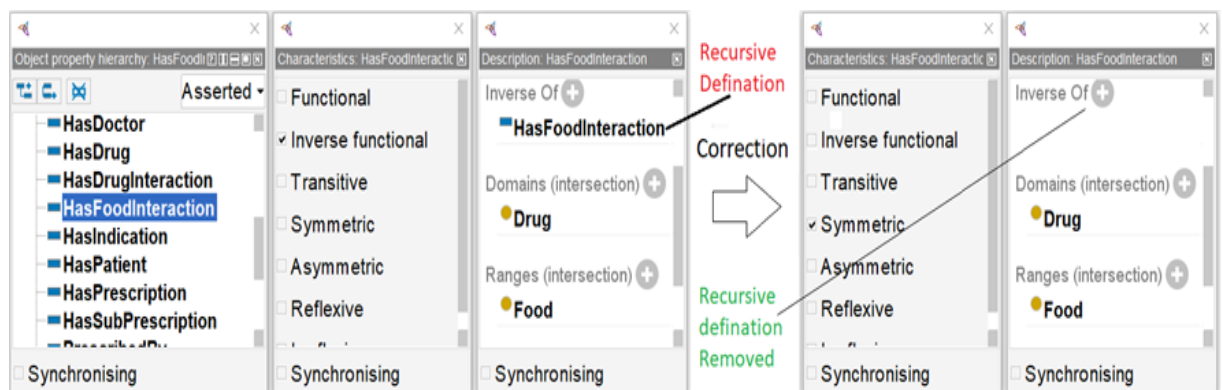


Figure 4-19 Correction of Pitfall P24 in protégé Removing recursion definition of self-inverse relation and adding symmetric property

4.6.3 Classification by Evaluation Criteria: Completeness Metric

Completeness evaluates whether the measure of the domain of interest is appropriately covered. The Completeness checkbox was selected under *Classification by Evaluation Criteria*, and the evaluation was run on the OOPS interface. The pitfalls to be checked are P04, P10, P11, P12, and P13. No bad practices for ontology completeness were detected in the ontology regarding pitfalls P04, P12, and P13, as shown in **Figure 4-21** Pitfalls P10 was detected due to missing disjointness in the ontology, and pitfall P11 was detected due to the missing domain and range properties.

Congratulations!
Your ontology does not contain any bad practice detectable by OOPS! from the ones you have chosen.

Figure 4-20 Results for pitfall P04, P12, and P13.

Results for P10: Missing disjointness.		ontology* Important
The ontology lacks disjoint axioms between classes or between properties that should be defined as disjoint.		
*This pitfall applies to the ontology in general instead of specific elements.		
Results for P11: Missing domain or range in properties.		11 cases Important
Object and/or datatype properties without domain or range (or none of them) are included in the ontology.		
• This pitfall appears in the following elements:		
> http://www.iepo.com/iepo#PatientDataProperties	> http://www.iepo.com/iepo#PrescriptionDataProperties	
> http://www.iepo.com/iepo#ClinicDataProperties	> http://www.iepo.com/iepo#AppointmentDataProperties	
> http://www.iepo.com/iepo#Instruction	> http://www.iepo.com/iepo#DiseaseDataProperties	
> http://www.iepo.com/iepo#PersonDataProperties	> http://www.iepo.com/iepo#AllergyDataProperties	
> http://www.iepo.com/iepo#FoodDataProperties	> http://www.iepo.com/iepo#PharmacyDataProperties	
> http://www.iepo.com/iepo#DrugDataProperties		

Figure 4-21 Pitfall P10 and P11 results expanded to show the affected ontology elements

Corrections were made to address pitfalls P10 and P11. For P10, all the data properties belonging to a particular class or having the same domain were disjointed to avoid interference. **Figure 4-22** shows an example of adding disjointness to all the Prescription class data properties.

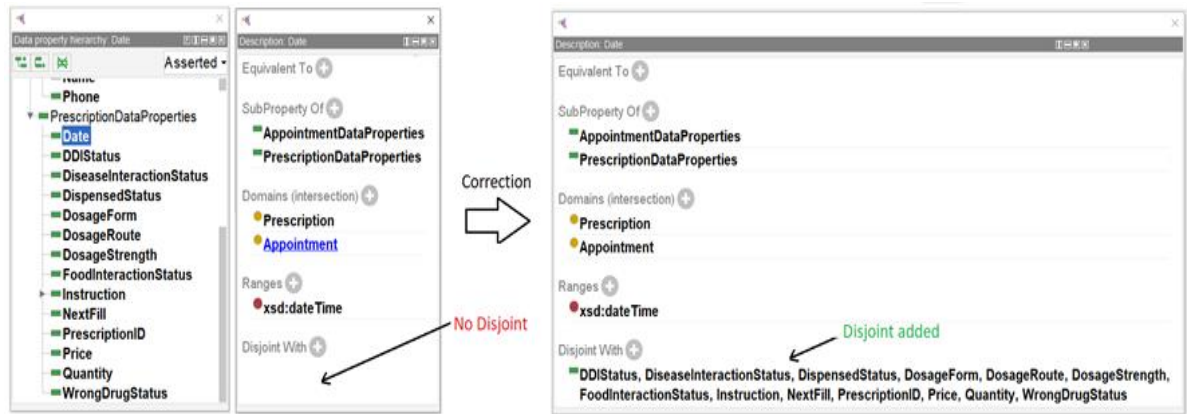


Figure 4-22 Correction of pitfall P10 in Protégé: Adding Missing Disjoints

The complaint by OOPS for P11 concerning missing domain and range will not be corrected because the 11 elements or the data properties in question are not used in the ontology but enables the grouping of data properties into their respective classes, which makes it easier to develop the ontology. Therefore, P11 will not be taken as a pitfall because the owl:topDataProperty in **Figure 4-20**, an inbuilt owl property is also a grouping data property with no domain and range, but OOPS does not detect it. All 11 cases for P11 refer to the parent grouping data properties under owl:topDataProperty where every class has its data properties grouped.

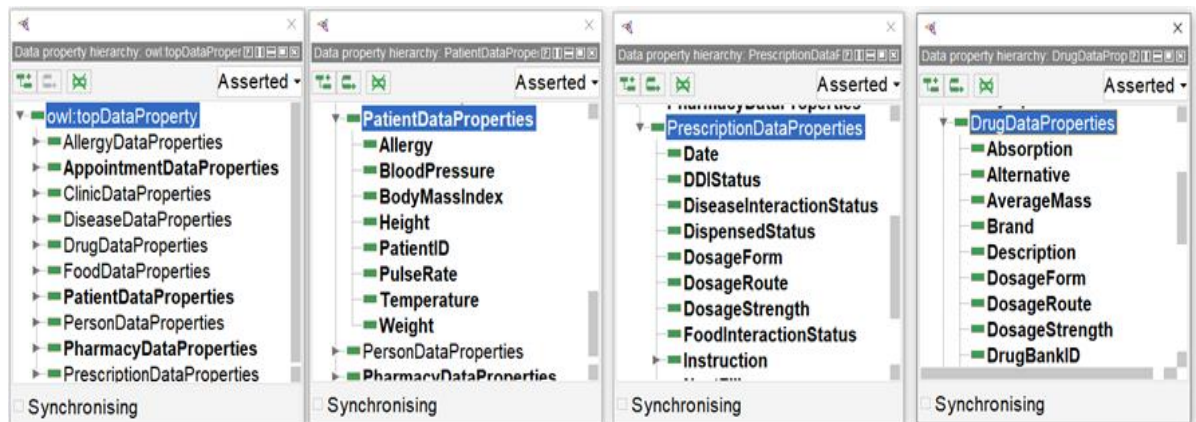


Figure 4-23 Parent Data Properties for Grouping Data Properties for Each Class.

It can be time-consuming to locate a particular data property when the data properties are many. To further validate our claim of dismissing P11 as a pitfall, let us consider

PatientDataProperties, one of the 11 cases for P11 which is missing a domain or range. The *PatientDataProperties* group the patient data properties for neatness and ease of locating them in protégé similar to *owl:topDataProperty*. Each class has its data property grouped onto the parent data property, which has no domain or range. As such, P11 will not be corrected but left as it is, which will not affect the ontology.

4.6.4 Classification by Evaluation Criteria: Conciseness Metric

Conciseness checks for irrelevant elements regarding the representation of the semantics being redundant. The pitfalls to be checked contributing to ontology conciseness metrics are P02, P03, and P21. **Figure 4-21** shows the results after running the OOPS evaluation scan. The results show that no pitfalls were detected in the ontology when checking for conciseness, meaning the ontology is 100% concise.

Congratulations!
Your ontology does not contain any bad practice detectable by OOPS! from the ones you have chosen.

Figure 4-24 Results for Conciseness Metric.

4.6.5 IEPO Evaluation using Classification by Dimension

The classification by dimension evaluates the ontology based on ten metrics divided into three dimensions: structural dimension, functional dimension, and usability-profiling dimension, as described in the following sections. Figure shows the classification of pitfalls according to the dimensions.

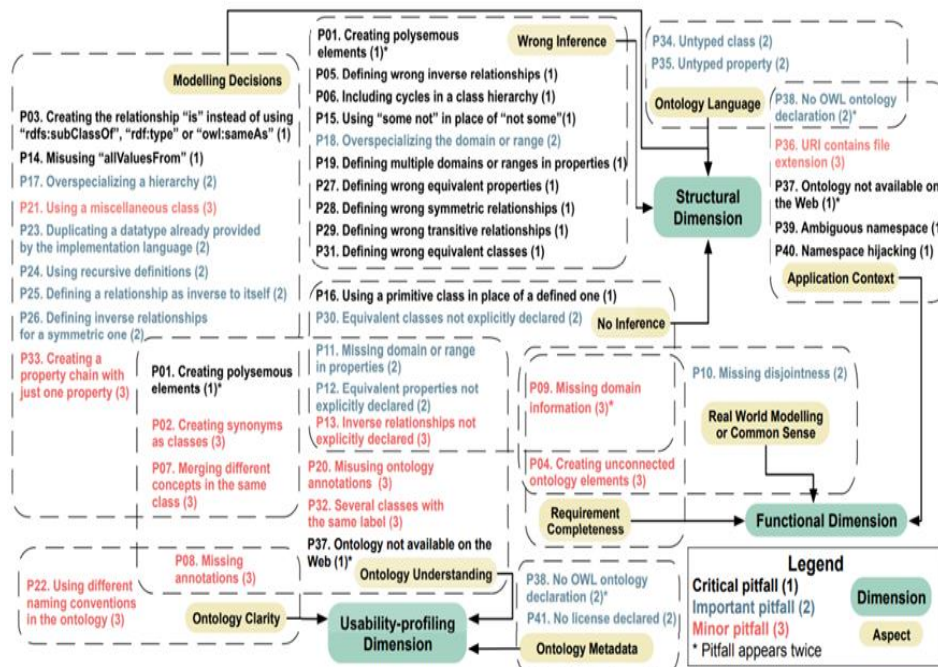


Figure 4-25 Pitfall Classifications by dimensions

4.6.5.1 Classification by Dimension: Structural Dimension Metric

The structural dimension focuses on syntax and formal semantics. Four metrics are evaluated under the structural dimension: modelling decisions, wrong inference, no inference, and ontology language. These metrics are described below.

4.6.5.2 Modelling Decisions

This aspect involves evaluating whether developers use the primitives provided by ontology implementation languages in a correct way and if there are modelling decisions that could be improved. Checks for pitfalls P02, P03, P07, P21, P24, P25, P26 and P33.

Congratulations!

Your ontology does not contain any bad practice detectable by OOPS! from the ones you have chosen.

Figure 4-26 Results for Modelling Decisions

4.6.5.3 *Wrong Inference*

This aspect refers to the evaluation of the inference of erroneous or invalid knowledge. Checks for pitfalls P05, P06, P19, P27, P28, P29 and P31

Results for P19: Defining multiple domains or ranges in properties.	11 cases Critical 
---	---

Figure 4-27 Results for Wrong Inference Metric

4.6.5.4 *No Inference*

This aspect refers to checking whether desirable or expected knowledge could be inferred from the given ontology, but it is not inferred. Checks for pitfalls P11, P12, P13 and P30.

Congratulations! Your ontology does not contain any bad practice detectable by OOPS! from the ones you have chosen.

Figure 4-28 Results for No Inference Metric

4.6.5.5 *Ontology language*

This aspect refers to checking whether the ontology is compliant both with the ontology language specification and with the syntax in which the ontology is formalized. Checks for pitfalls P34, P35 and P38.

Congratulations! Your ontology does not contain any bad practice detectable by OOPS! from the ones you have chosen.

Figure 4-29 Results for Ontology language Metric

4.6.5.6 *Ontology language:*

This aspect refers to checking whether the ontology is compliant both with the ontology language specification and with the syntax in which the ontology is formalized. Checks for pitfalls P34, P35 and P38.

Congratulations!
Your ontology does not contain any bad practice detectable by OOPS! from the ones you have chosen.

Figure 4-30 Results for Ontology language Metric

4.6.6 Classification by Dimension: Functional Dimension Metric

4.6.6.1 *Functional dimension:*

Focuses on the ontology conceptualization relating to the intended use of a particular ontology and its components. Three metrics are evaluated which contribute to the structural dimension: real-world modelling or common sense, requirements completeness and application context. These metrics are described below.

4.6.6.2 *Real-World Modelling or Common Sense:*

This aspect deals with the knowledge that domain experts expect to appear in the ontology, but it is not represented. Checks for pitfall P04 and P10.

Congratulations!
Your ontology does not contain any bad practice detectable by OOPS! from the ones you have chosen.

Figure 4-31 Results for Real World Modelling Metric

4.6.6.3 *Requirements Completeness:*

This aspect deals with the coverage of the requirements specified in the Ontology Requirements Specification Document. Checks for pitfall P04 and P09.

Congratulations!
Your ontology does not contain any bad practice detectable by OOPS! from the ones you have chosen.

Figure 4-32 Results for Requirements Completeness Metric

4.6.6.4 *Application context:*

This aspect refers to the adequacy of the ontology for a given application or use case. Checks for pitfalls P36, P37, P38, P39, and P40.

Congratulations!
Your ontology does not contain any bad practice detectable by OOPS! from the ones you have chosen.

Figure 4-33 Results for Application Context Metric

4.6.7 Classification by Dimension: Usability-Profiling Dimension Metric

4.6.7.1 *Usability-Profiling:*

Deals with an ontology's communication context which is the profile of the ontology in terms of annotations. The metrics that contribute to the structural dimension are ontology clarity, ontology understanding, and ontology metadata. These metrics are described below.

4.6.7.2 Ontology Clarity:

This aspect refers to the properties of ontology elements of being easily recognizable and understood by the user. Checks for pitfalls P08 and P22. **Figure 4-34** shows the results for the ontology clarity metric and P08 was detected with 99 minor cases due to a lack of annotation of all classes, subclasses, data properties, and object properties in the ontology. Annotations are human descriptions of the ontology concepts and elements for easier understanding. With no annotations, the ontology has no clear clarity.

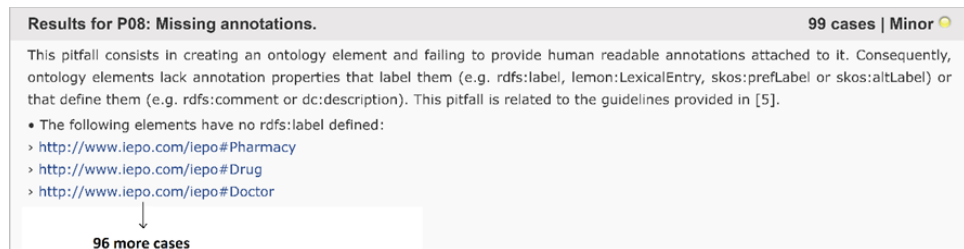


Figure 4-34 Results for Ontology Clarity Metric

For successful ontology clarity, pitfall P08 needs to be corrected. Protégé has built-in `rdfs` annotations. The *label* and the *comment* annotations were attached to the 99 elements in P08. The label annotation describes the unique name of the element, and the comment is the description of the concept taken from the glossary of terms section under conceptualization. The missing annotations were added to all the 99 elements comprising classes, subclasses, data properties, and object properties.

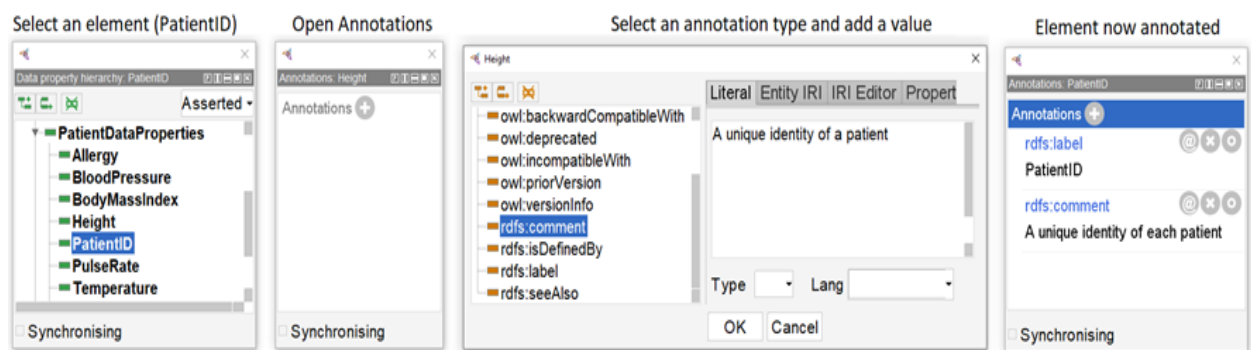


Figure 4-35 Correction of pitfall P08 for ontology clarity: Adding annotations

Ontology Understanding: This aspect involves evaluating any kind of information that can help the user to understand the ontology content. Checks for pitfalls P02, P07, P08, P11, P12, P13, P20, P32, and P37

Congratulations!
Your ontology does not contain any bad practice detectable by OOPS! from the ones you have chosen.

Figure 4-36: Results for Real World Modelling Metric

Ontology Metadata: This aspect involves evaluating the existence of information that can help to understand the ontology context itself, instead of the conceptualization defined. Some examples of this information are licensing, provenance, versioning, etc. Checks for pitfalls P38 and P41

The ontology metadata is mainly checking if the ontology has a license. An ontology license protects an intellectual creation by copyright, so practical use will require permission. Figure 4-37 shows the results for ontology metadata pitfall checking, asserting that the ontology has no license.

Results for P41: No license declared. ontology* | Important

The ontology metadata omits information about the license that applies to the ontology.

- This pitfall appears in the following elements:
 - > It has no licence defined

Figure 4-37 Results for Ontology Metadata Metric

As shown in Figure 4-38, pitfall P41 was corrected by adding an ontology license to IEPO in Protégé. The license was created at the Creative Commons website (<https://creativecommons.org/>) and was made open source, allowing even commercial use of the ontology for free. The label, comment, license, and rights annotations were used to describe the ontology metadata in the ontology header, with the license and rights annotation describing the license.

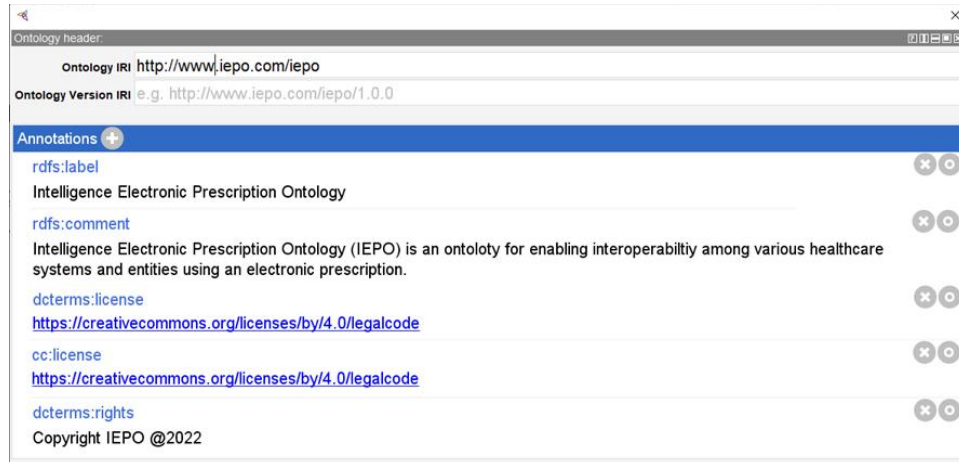


Figure 4-38 Correction of pitfall P08 for Ontology Metadata Metric: Adding license

4.6.8 Summary of OOPS ontology evaluation

After correcting all the pitfalls in Protégé, the final OOPS evaluation gave successful results with no pitfalls detected. The ontology passed all the ontology metrics, as shown in Table 1 and Table 2. The percentage of pitfalls before correction in the tables shows the percentage error of the evaluated ontology on that specific ontology metric selected. For example, the consistency evaluation metric yield 60% of detected pitfalls, meaning that the ontology is $100 - 60 = 40\%$ consistent before correcting the pitfalls in Protégé. The same was applied to all the metrics with $100\% \text{ minus (Percentage of pitfalls before correction) as the ontology's metric percentage level in that metric}$, e.g., the Ontology has $100\% - 0\% = 100\%$ Consciousness. The last column in the tables shows the percentage of pitfalls detected after correcting all the detected pitfalls in the ontology with Protégé and then re-evaluating the ontology in OOPS. The results show that all the pitfalls were eliminated with 0% of pitfalls detected, giving a 100% metric percentage level in all the metrics.

Table 4-3 OPPS Evaluation Summary of Classification by Evaluation Criteria

Ontology Metric to be Selected	OOPS Pitfalls to be Checked	No. OOPS Pitfalls detected	Percentage of pitfalls before correction	Percentage of pitfalls detected after correction
Consistency	P05, P06, P07, P19, P24.	P05, P19, P24	3/5 = 60%	0%
Completeness	P04, P10, P11, P12, P13	P10, P11	2/5 = 40%	0%
Conciseness	P02, P03, P21	Non	0/3 = 0%	0%

Table 4-4 OPPS Evaluation Summary of Classification by Dimension

Ontology Metric to be Evaluated	OOPS Pitfalls to be Checked	OOPS Pitfalls detected by evaluation	Percentage of pitfalls before correction	Percentage of pitfalls detected after correction
Structural Dimension				
Modelling Decisions	P05, P06, P07, P19, P24.	P05, P19, P24	3/5 = 60%	0%
Wrong Inference	P04, P10, P11, P12, P13	P10, P11	2/5 = 40%	0%
No Inference	P02, P03, P21	Non	0/3 = 0%	0%
Ontology language	P34, P35, P38	Non	0/3 = 0%	0%
Functional Dimension				
Real World Modelling	P05, P06, P07, P19, P24.	P05, P19, P24	3/5 = 60%	0%
Requirements Completeness	P04, P09	Non	0/3 = 0%	0%
Application context	P04, P10, P11, P12, P13	P10, P11	2/5 = 40%	0%
Usability-Profiling Dimension				
Ontology Clarity	P08 and P22	P08	1/2 = 50%	0%
Ontology Understanding	P02, P07, P08, P11, P12, P13, P20, P32, P37	P08, P11	2/9 = 22%	0%

Ontology Metadata	P38, P41	P41	1/2 = 50%	0%
-------------------	----------	-----	-----------	----

4.7 IEPO Interoperability Metrics and Prototype

Healthcare interoperability will be demonstrated by several metrics as follows. The justification of the IEPO Interoperability metrics and prototype is to ascertain the communication and share information between the two different healthcare systems of both systems.

4.7.1 Competency Questions Metric:

Competency questions were presented in Section 3.4 as ontology specifications check whether an ontology gives the correct answers when queried using SPARQL. The competency questions were converted to SPARQL queries to query the IEPO server to demonstrate interoperability between systems by returning correct results as answers from the IEPO server. A PHP prototype was designed to render the SPARQL query results in a user-friendly GUI. The results of competency questions are presented in Section 4.4.3.

4.7.2 IEPO Prototype:

IEPO prototype is a centralized prescribing web application where all healthcare entities can share healthcare information securely anywhere at any time. GUI users have roles like a doctor, patients, pharmacists, etc., and the GUI renders pages according to the role. A patient has access to only their prescription information. Only doctors have a "Prescribe Medication" page. The prototype GUI is presented in Section 4.4.3.

4.7.3 Security: The IEPO prototype

Security is by access control using a username and password. Data encryption of the healthcare data in the IEPO server is crucial and will be presented in future works.

4.7.4 Novel Interoperability Index Metric (IA→B):

This research proposes a novel metric called interoperability index denoted IA→B to quantify the interoperability strength or degree between system A and system B when system A requests information from system B.

$$I_{A \rightarrow B} = \frac{|A \cap IEPO|}{|A|}, I_{A \rightarrow B} \in [0, 1] \quad (8)$$

Where A is system A, and B is system B to be interoperable. The index IA→B shows the interoperability strength from system A to system B. If system A requests information from system B, what is the ratio of all the concepts that system A requires from system B? With IEPO, system A will request system B's information not from B directly but from the IEPO server. The expression $|A \cap IEPO|$ is the number of common concepts in systems A and IEPO. The higher the value of IA→B, the higher the degree of interoperability. The term $|A| \cap |IEPO|$ is the number of concepts (information) common to system A and IEPO. In IEPO, all system concepts are integrated into a single ontology. Every system will have all its concepts in the IEPO so that the expression $|A| \cap |IEPO|$ for every system in IEPO will yield the value $|A|$. The same applies to the interoperability index IB→A from system B to system A denoted as:

$$I_{B \rightarrow A} = \frac{|B \cap IEPO|}{|B|}, I_{B \rightarrow A} \in [0, 1] \quad (9)$$

4.7.5 Interoperability Response Time

This is the time it takes for two systems to interoperate or exchange information in milliseconds. In IEPO, response time is the time taken for executing SPARQL queries like searching, prescribing, and recommending prescriptions between a user (denoted by the GUI) and the

IEPO server. The calculations of this time are shown in Figure 9 using the PHP micro time function before and after the SPARQL query to capture the start time and finish time, respectively.

4.7.6 IEPO Interoperability Results and Discussion

4.7.7 IEPO Prototype and Competency Questions

Figure 4-39 shows the GUI of the proposed IEPO framework that demonstrates interoperability with a secure login-based web application. The PHP application is hosted on a XAMMP server on Windows 10, 8GB PC. The GUI in **Figure 4-39** was also accessed using an iPad Pro connected to the same WiFi network as the IEPO XAMPP server with IP address 10.12.21.34, using Chrome browser. The server's IP address was identified by the running ipconfig command on the command console (CMD). Every entity or user can log in and access the specific information required via the internet. This web application is platform-independent and requires no installation as it can run in any web browser. Users can create, search, read, update, recommend, and delete prescriptions according to their roles. Not all competency questions will be shown as they will take up a lot of space.

The screenshot displays the IEPO ePrescription APP interface. At the top, a light blue header contains the text "Hi, eunice. Welcome to IEPO ePrescription APP." and two buttons: "Reset Your Password" (yellow) and "Sign Out of Your Account" (red). Below this is a dark navigation bar with links: Home, Search Prescription (highlighted in green), View All Prescriptions, Prescribe Medication, Medical History, Change Prescription, Drug Recommendation (Pharmacist), Drug Info, Disease Info, and Allegies.

Below the navigation bar is a search input field containing "P0000001" and a magnifying glass icon. Below this is a green banner with the text "Search Results for **Docus Hope** of Patient ID: **P0000001**".

The main content area is a light blue box containing the following prescription details:

- PATIENT ID:** P0000001
- FULL NAME:** Docus Hope
- PRESCRIPTION ID:** Docus_Hope_002
- PRESCRIPTION DATE:** 29 June 2022
- DDI STATUS:** *DDI Exists (Metoclopramide Has DDI with Risperidone in Current Prescription with ID: Docus_Hope_002)*
- WRONG DRUG STATUS:** *Wrong Drug (Metoclopramide Indication is not HIV-1 but Diabetic Gastroparesis)*
- MEDICATION:** Metoclopramide
- SICKNESS:** HIV-1
- DOSAGE:** 10mg
- INSTRUCTIONS:** 1 tablet 4 times a day, taken 30 minutes before each meal and at bedtime each meal and at bedtime
- QUANTITY:** 120 tablets
- PRICE:** US\$ 20.00 for 120 tablets
- REFILL:** 1 month
- PRESCRIBER:** Dr Davis
- DISPENSE STATUS:** *Not Yet*
- DISPENSED BY:** N/A
- DATE DISPENSED:** N/A
- PHARMACY NAME:** Midnight Pharmacy
- PHARMACY ADDRESS:** Grouve Road, Plot 448,Botswana, Tel +267 429 0000

Figure 4-39 The IEPO E-Prescription Prototype

The query is supplied with the patient ID "P0000001" in the IEPO GUI search box and returns correct results rendered to the GUI as shown in **Figure 4-39**. Correct results mean the competency questions were answered successfully, and the GUI user, either doctor, patient, pharmacist, etc., can interoperate with the system and get the required results. **Figure 4-40** shows the query for recording the Patient's sickness as HIV during prescribing. All prescription attributes are captured from the GUI field during prescribing and successfully

written to the IEPO Triple database using this SPARQL INSERT query. The prescription details are then rendered to the GUI when searched using Patient ID in **Figure 4-39**.

Pharmacy (A) -> Clinic (B) Interoperability Index:

The pharmacy system will be denoted as system A and the clinic as B. The interoperability degree from the pharmacy to the clinic will then be denoted $I_{A \rightarrow B}$. The pharmacy system in Section 3.4.1 in Figure 3 has 13 main concepts, $A = 13$, IEPO has 31 main concepts, $IEPO = 31$, while

```
<?php
//Query execution start time
$time_start = microtime(true);

$search = "Miracle_Name3";
$patient = "<http://www.iepo.com/iepo#".$search.">";
echo htmlspecialchars($patient);
$result = $sparql->update(
    'INSERT DATA {'.
    '    '.$patient.' <http://www.iepo.com/#Has_Sickness>'.
    '    <http://www.iepo.com/iepo#AIDS> .'.
    '}'
);
//Query execution end time
$time_end = microtime(true);

echo "<br> IEPO update response time is:".
(($time_end-$time_start)/1000)." sec<br>";
echo "<br>Query updated successfully";
?>
```

Figure 4-40 IEPO SPARQL INSERT Query for recording the Patient's sickness as HIV.

Figure 4-40 shows the IEPO SPARQL INSERT Query for recording the Patient's sickness as HIV during prescribing. The clinic system has 18 concepts, $B = 18$. Then:

$$I_{A \rightarrow B} = \frac{|A| \cap |IEPO|}{|A|} = \frac{13}{13} = 100\% \quad (10)$$

The results show that the pharmacy is 100% interoperable with the clinic, meaning that all the information that the pharmacy requires from the clinic is made available via IEPO.

Clinic (B)-> Pharmacy (A) Interoperability Index: The interoperability index from Clinic to Pharmacy system is denoted $I_{B \rightarrow A}$:

$$I_{B \rightarrow A} = \frac{|B \cap IEPO|}{|B|} = \frac{18}{18} = 100\% \quad (11)$$

These results show that the clinic system is 100% interoperable with the pharmacy system, meaning all the information that the clinic requires or requests from the pharmacy is made available via IEPO. Comparing IEPO with when clinic and pharmacy are stand-alone and not centralized, then from Figure 3, common concepts between the clinic and the pharmacy system are 11. Then $I_{A \rightarrow B}$ can be calculated as:

$$I_{A \rightarrow B} = \frac{|A \cap B|}{|A|} = \frac{11}{13} = 86\% \quad (12)$$

The 86% means that 14% of concepts in the pharmacy cannot be obtained from the clinic. Interoperability from clinic to pharmacy with no IEPO centralized access can be calculated as:

$$I_{B \rightarrow A} = \frac{|B \cap A|}{|B|} = \frac{11}{18} = 61\% \quad (13)$$

The 61% means 61% of the concepts in the clinic cannot be accessed from the pharmacy. It can be asserted that IEPO achieves 100% interoperability between systems, while in stand-alone systems if they are to be interoperable, they can achieve between 61% and 86% interoperability degrees.

4.7.8 The Interoperability Response Time of IEPO

Table 4-5 The interoperability Response Time of IEPO

Response Time (ms)	SPARQL Queries Executed between GUI and IEPO Server					
	Retrieve Prescription	Prescribe Prescription	Drug Recommendation	Detect DDI Error 1	Detect DDI Error 2	Detect Wrong Drug
IEPO	0.026	0.26	0.586	0.155	0.093	0.124
Google Recommendation	200	200	200	200	200	200

Table 4-5 above shows the interoperability response time (ms) of six important IEPO requests to be performed by various health actors between the GUI and the IEPO server. The recommended Google response time between the user and the server should not exceed 200ms [55]. It can be seen in **Table 4-5** that for all six queries, IEPO has a swift response compared to the recommended value meaning that results are rendered and written to the

server at almost negligible speeds. The drug recommendation query has the most significant response time of 0.586ms because it involves the pharmacy sending the request to the doctor via IEPO. The doctor then performs some error checking on the new drug to be prescribed, and then performs an update query to the IEPO, which is similar to the prescribing query that writes data to the IEPO server. It can be asserted that IEPO has a maximum response time of 0.586ms which is 343% better than the recommended 200ms Google response time.

4.8 Competency Questions Evaluation

The competency questions are also SPARQL queries and have been demonstrated during the vital signs health monitoring section and the analytical section of the results. This section will be an additional section of competency questions. The competency question also demonstrates that IEPO is interoperable by sharing information amongst the entities. The competency questions are queries on the IEPO framework that should give exact results. These queries are what a doctor, a patient, or a pharmacist can ask to get information from the IEPO system remotely. The queries are performed via the internet through the IEPO GUI or an IEPO SPARQL endpoint in the Jena Fuseki Query Interface. The same SPARQL queries can also be performed locally in the Protégé tool at the development stage, but the results will be similar. A doctor can request information about the patient, the pharmacy, the drugs etc. A patient can request information about his prescription history. A pharmacist can make recommendations to the prescriber for alternative drugs for a particular patient due to shortage or cost. In this research, the competency question will be converted to SPARQL queries and then the queries being executed in Protégé “SPARQL QUERY TAB” and output the results. The competency question of statistical analysis to analyze the data, medical error healthcare monitoring will be demonstrated. Competency questions demonstrate interoperability between health systems because any health entity in IEPO can access information for all other entities depending on the roles and restrictions set by the administrator e.g., a doctor can get information about all his patient’s prescriptions even those that were prescribed by other doctors. If access control is stripped off in IEPO, any user can

interoperate with all health systems registered with IEPO and can get any information they need from the integrated systems.

Figure 4-41 shows the SPARQL SELECT query for Competency Question "What are the prescription details for a patient with patientID P0000001?".

```
/* Query to search for patient prescription*/
$patientID = "P0000001"
$query = '
PREFIX iepouri: <http://www.iepo.com/iepo#>
SELECT ?patient ?p ?id ?fn ?pid ?pdate ?ddis
      ?wds ?drug ?s ?dsg ?instr ?qt ?pr ?rf
      ?prcb ?dst ?dsb ?ddate ?pn ?padd
WHERE { ?patient iepouri:HasPrescription ?p;
        patient iepouri:PatientID . '$patientID'.
        iepouri:FullName ?fn;
        ?p iepouri:PrescriptionID ?pid;
        iepouri:PrescriptionDate ?pdate;
        iepouri:DDISStatus ?ddis;
        iepouri:WrongDrugStatus ?wds;
        iepouri:Medication ?drug
        iepouri:Sickness ?s;
        iepouri:Dosage ?dsg;
        iepouri:Instructions ?instr;
        iepouri:Quantity ?qt;
        iepouri:Price ?pr;
        iepouri:Refill ?rf;
        iepouri:Prescriber ?prcb;
        iepouri:DispenseStatus ?dst;
        iepouri:DispensedBy ?dsb;
        iepouri:DateDispensed ?ddate;
        iepouri:PharmacyName ?pn;
        iepouri:PharmacyAddress ?padd;
      }
ORDER BY ?p';
```

Figure 4-41 IEPO prescription retrieval SPARQL Query for Competency Question 1

4.9 Other IEPO Ontology Evaluation Metrics:

The proposed IEPO ontology is further evaluated with quantifiable ontology metrics as follows.

1. Basic IEPO Ontology Metrics: Figure below shows the basic ontology metrics. These metrics were accessed in Protégé software by going to Window > Views > Ontology views> Ontology metrics.

Metrics	
Axiom	546
Logical axiom count	342
Declaration axioms count	190
Class count	38
Object property count	39
Data property count	27
Individual count	85
Annotation Property count	6
Class axioms	
SubClassOf	34
☒ Synchronising	

Figure 4-42 Basic IEPO ontology Metrics from Protégé.

2. Ontology Size: Size refers to the number of entities in an ontology, $|M|$. This can be further subdivided into class size $|C|$, object property size $|OP|$, data property size $|DP|$, and individual size $|I|$.

$$\text{Size}(M) = |M| = |C| + |OP| + |DP| + |I| = 38 + 39 + 27 + 85 = 189$$

The Ontology Size of 189 is 75% closer to the optimum value of 250 ontology size according to [57].

CHAPTER 5 CONCLUSION AND RECOMMENDATION

5.0 Introduction

A brief summary of the work done, results obtained and findings are reported in this chapter. Finally, recommendations leading to further research are also highlighted.

5.1 Conclusion of Dissertation based on the objectives.

This research, as defined by the objectives, focused on developing an Intelligent e-prescription Ontology (IPO) framework for healthcare interoperability using semantic e-prescription. A case study of Interoperability between a doctor and a pharmacist to prevent specifically Drug-to-Drug Interaction (DDI) and wrong drug medical errors were considered using the An important advantage of this framework is that many healthcare entities like a doctor, pharmacists, medical aid, research, and machines can have access to all patient's prescriptions. The followings are how the objective stated in Section 1.4.2 was achieved.

OBJECTIVE 1: To conduct a comprehensive literature review on semantic e-prescription interoperability frameworks to determine the existing gaps.

This objective was archived in Section 2.6 where several eprescription interoperability frameworks from the literature were presented, analyzed, and the gaps identified.

OBJECTIVE 2: To conduct a comprehensive literature review to find the most suitable ontology development methodologies, tools, languages, and suitable ontology evaluation methodologies, tools, and metrics to adopt for IEPO development.

This objective was archived in Section 2.5.

OBJECTIVE 3: To create an electronic prescription ontology called IEPO for healthcare interoperability.

This objective was archived in Chapter 3 where an open-source ontology for healthcare interoperability was developed and made available for download online.

OBJECTIVE 4: To create a SPARQL endpoint Application Programming Interface (API) to enable various healthcare stakeholders like doctors, pharmacists, and patients to access IEPO healthcare data seamlessly and remotely via the internet.

This objective was also archived in Chapter 3 at the implementation phase of methontology, where a prototype was developed to demonstrated interoperability. The prototype had a GUI where doctors, pharmacists, and patients can interact and share data.

OBJECTIVE 5: To evaluate the performance of the proposed IEPO framework.

This objective was archived in Chapter 4, where the IEPO framework was evaluated in terms of:

1. The Ontology Domain Knowledge Coverage Evaluation: The performance of the proposed framework in terms of domain knowledge coverage was analyzed and compared to existing works. The results showed that, on average, IEPO is the best e-prescription ontology currently, which performs 76% better than existing ePrescription ontologies regarding ePrescription domain knowledge coverage.
2. Medical Error Detection Evaluation: To detect medical errors like the DDI, wrong prescription, etc., for a particular disease using the SWRL. It was discovered that IEPO has 100% accuracy in terms of medical error detection outperforming related works significantly.
3. Vital signs healthcare monitoring: The proposed IEPO framework was able to give abnormalities where a vital sign is out of the normal range, like COVID-19 alerts for abnormal temperatures.
4. Prescription Analysis: The proposed IEPO framework accurately analyzed the IEPO data to give analytical information using SPARQL queries.
5. Ontology Evaluation using Ontology Pitfall Scanner: To check for the quality, correctness, and structure of the IEPO using the detailed custom metric with the categorized evaluation metric. The results have shown that IEPO, after evaluation with the OOPS online pitfall scanner, yielded zero errors regarding various ontology metrics like consistency, completeness, conciseness, etc.
6. IEPO Interoperability Metrics and prototype: IEPO demonstrated high interoperability index of 100% between the clinic and the pharmacy compared to when they are stand-alone.
7. Usability Rating: The IEPO framework is 12% more usable than related works, with an 87% usability score rating.

8. Ontology Competency Questions using SPARQL: To further test if the ontology worked and was able to answer questions using the SPARQL. IEPO was able to answer the competency questions accurately, which were implemented using a SPARQL query language.

5.2 ANSWERS TO RESEARCH QUESTIONS

RESEARCH QUESTION 1: What are the gaps found in current healthcare interoperability frameworks from literature based on semantic e-prescription to be addressed by the proposed IEPO framework?

- Lack of interoperability: Cost, time, Loss of lives.
- No medical error detection
- No formal ontology methodology followed
- No framework evaluation
- No implementation
- Not open source
- Ontology not made publicly available and downloadable online
- No implementation Prototype or GUI
- Poor domain knowledge Coverage.
- No pharmacy recommendations
- No medical error detection
- No formal ontology methodology followed
- No framework evaluation
- Not open source
- Ontology not made publicly available and downloadable online
- No implementation Prototype or GUI
- Poor domain knowledge Coverage.
- No pharmacy recommendations

RESEARCH QUESTION 2: What are the most suitable healthcare interoperability types and approaches to adopt for developing the IEPO framework?

- Adopted Interoperability Type: Semantic Interoperability with justification in the introduction section
- Adopted Interoperability Approach: Centralized Approach with justification in the introduction section

RESEARCH QUESTION 3: What are the most suitable interoperability standards to adopt for the development of IEPO framework?

- Adopted Healthcare Interoperability Standard: SNOMED with justification in the introduction section

RESEARCH QUESTION 4: Between ontologies and databases, which is the most suitable to adopt for developing the IEPO framework?

- Ontologies with justification in the literature review section

RESEARCH QUESTION 5: What are the most suitable ontology development methodologies, languages and tools to adopt for the development of IEPO framework?

- Adopted Ontology Development Methodology from Literature Review: Methontology with justification in the literature review section
- Adopted Ontology Development Languages from Literature Review: OWL2 with justification in the literature review section.
- Adopted Ontology Development Tools from Literature Review: Protégé 5.5.0 with justification in the literature review section

RESEARCH QUESTION 6: What are the most suitable ontology evaluation methodologies, metrics, tools and languages to adopt for evaluating the proposed IEPO framework?

- Adopted Ontology Evaluation Methodologies: Corpus-Based, OOPS, and Competency Questions,
- Adopted Ontology Evaluation Metrics: Domain Knowledge, consistency, completeness, correctness, and conscience etc.

- Adopted Ontology Evaluation Tools: OOPS
- Adopted Ontology Evaluation Languages: SPARQL

RESEARCH QUESTION 7: How can ontologies enable healthcare interoperability in the proposed framework?

- From the chapter 3 and chapter 4, ontologies enabled healthcare interoperability through the use of electronic semantic prescription. Ontologies standardize the concepts and meaning of the healthcare systems to be interoperable so that they can understand each other. Ontologies enabled healthcare interoperability between the clinic and the pharmacy healthcare system by creating ontologies for both systems and then merging the ontologies to create one ontology called IEPO.owl. The IEPO ontology was hosted on an online Jena Fuseki server to be queried via a GUI or a SPARQL endpoint, which enable both doctors, pharmacies and patient to access data for both systems like prescription medical history.

RESEARCH QUESTION 8: Will the proposed ontology be open source and be made available online for free download?

- The IEPO ontology was made open source and also made available for free download online @ shorturl.at/dqGY1

RESEARCH QUESTION 9: How will doctors, patients, and pharmacist share information with each other in the proposed framework?

- The proposed IEPO framework disseminated data to doctors, patients, and pharmacists using a SPARQL endpoint API and GUI. Doctors, patients, and pharmacists can query the endpoint with competency questions and access the GUI to prescribe, view, search, and recommend prescriptions in the IEPO framework seamlessly.

RESEARCH QUESTION 10: How well does the proposed IEPO framework perform compared to other related works?

- The proposed IEPO framework outperforms related works significantly as seen in the results section in terms of domain interoperability, domain knowledge coverage, medical error detection, vital signs monitoring and other metrics.

5.3 Further Research Recommendations

Finally, though the proposed IEPO framework outperforms the existing e-prescription ontology frameworks, further research work will be needed in the future to enhance healthcare interoperability. In the future, Security needs to be improved like eprescription data encryption since multiple users can query the SPARQL Endpoint; Also, it will be necessary to test the IEPO error detection performance with thousands of sample prescriptions for scalability and execution time. In addition, IEPO will be extended to cover more prescription errors like detection of wrong dosage, wrong route, etc. In addition, more evaluation metrics will be implemented to further check ontology's quality and correctness. The corpus evaluation methodology will be improved to include more image samples from Bing and Yahoo. The 1.4GB Drugbank XML drug information file will be converted into a corpus and fed into the corpus evaluation methodology. Hosting the IEPO ontology on a permanent online server will be a future recommendation also.

References

- [1] D. S. Wishart *et al.*, “DrugBank 5.0: A major update to the DrugBank database for 2018,” *Nucleic Acids Res*, vol. 46, no. D1, pp. D1074–D1082, Jan. 2018, doi: 10.1093/nar/gkx1037.
- [2] Rayhan A. Tariq; Rishik Vashisht; Ankur Sinha; Yevgeniya Scherbak., “Medication Dispensing Errors And Prevention,” *Europe PMC*, 2022.
- [3] *To Err Is Human*. Washington, D.C.: National Academies Press, 2000. doi: 10.17226/9728.
- [4] “Advancing the responsible use of medicines Applying levers for change.” [Online]. Available: <http://ssrn.com/abstract=2222541>
- [5] R. Javed, T. Saba, S. Humdullah, N. S. Mohd Jamail, and M. Javed Awan, “An Efficient Pattern Recognition Based Method for Drug-Drug Interaction Diagnosis,” in *2021 1st International Conference on Artificial Intelligence and Data Analytics, CAIDA 2021*, Apr. 2021, pp. 221–226. doi: 10.1109/CAIDA51941.2021.9425062.
- [6] J. L. Montastruc *et al.*, “Fatal adverse drug reactions: A worldwide perspective in the World Health Organization pharmacovigilance database,” *Br J Clin Pharmacol*, vol. 87, no. 11, pp. 4334–4340, Nov. 2021, doi: 10.1111/bcp.14851.
- [7] S. Zakharov, N. Tomas, and D. Pelclova, “Medication errorsan enduring problem for children and elderly patients,” *Ups J Med Sci*, vol. 117, no. 3, pp. 309–317, Aug. 2012, doi: 10.3109/03009734.2012.659771.
- [8] L. Puustjärvi, “The Challenges of Electronic Prescription Systems Based on Semantic Web Technologies.”
- [9] D. Gupta and S. Malik, “Adopting Semantic Interoperability for Improved Healthcare.” [Online]. Available: <https://ssrn.com/abstract=4096399>
- [10] O. Iroju, A. Soriyan, I. Gambo, and J. Olaleke, “Interoperability in Healthcare: Benefits, Challenges and Resolutions,” 2013. [Online]. Available: <http://www.issr-journals.org/ijias/>

- [11] M. Lehne, J. Sass, A. Essenwanger, J. Schepers, and S. Thun, "Why digital medicine depends on interoperability," *npj Digital Medicine*, vol. 2, no. 1. Nature Publishing Group, Dec. 01, 2019. doi: 10.1038/s41746-019-0158-1.
- [12] RashmiBurse MichelaBertolotto DympnaO'Sullivan GavinMcArdle, "Semantic interoperability: the future of healthcare," *Web semantic*, pp. 31–53, 2021.
- [13] P. Machanick Paula Kotze *et al.*, "Editor-in-Chief Co-Editor Associate Editors: Information Systems Associate Editors: Computer Science Editorial Board Administration and support administration production," *South African Computer Journal Number*, vol. 50, 2013, [Online]. Available: <http://sacj.cs.uct.ac.za>
- [14] M. Robkin, S. Weininger, B. Preciado, and J. Goldman, "Levels of conceptual interoperability model for healthcare framework for safe medical device interoperability," in *ISPCE 2015 - IEEE Symposium on Product Compliance Engineering, Proceedings*, Jun. 2015. doi: 10.1109/ISPCE.2015.7138703.
- [15] A. Ahmad, S. Cuomo, W. Wu, and G. Jeon, "Intelligent algorithms and standards for interoperability in Internet of Things," *Future Generation Computer Systems*, vol. 92. Elsevier B.V., pp. 1187–1191, Mar. 01, 2019. doi: 10.1016/j.future.2018.11.015.
- [16] S. Bhartiya, D. Mehrotra, and A. Girdhar, "Issues in Achieving Complete Interoperability while Sharing Electronic Health Records," in *Physics Procedia*, 2016, vol. 78, pp. 192–198. doi: 10.1016/j.procs.2016.02.033.
- [17] T. Benson Grahame Grieve, "Principles of Health Interoperability." [Online]. Available: <http://www.springer.com/series/10471>
- [18] A. Boussadi and E. Zapletal, "A Fast Healthcare Interoperability Resources (FHIR) layer implemented over i2b2," *BMC Med Inform Decis Mak*, vol. 17, no. 1, Aug. 2017, doi: 10.1186/s12911-017-0513-6.
- [19] B. M. H. K. G. G. & P. B. Suranga N. Kasthurirathne, "Enabling Better Interoperability for HealthCare: Lessons in Developing a Standards Based Application Programing Interface for Electronic Medical Record Systems," *Journal of Medical Systems*, 2015.

- [20] C. Gaudet-Blavignac, V. Foufi, M. Bjelogric, and C. Lovis, "Use of the systematized nomenclature of medicine clinical terms (snomed ct) for processing free text in health care: Systematic scoping review," *Journal of Medical Internet Research*, vol. 23, no. 1. JMIR Publications Inc., Jan. 01, 2021. doi: 10.2196/24594.
- [21] M. Stram *et al.*, "Logical observation identifiers names and codes for laboratorians potential solutions and challenges for interoperability," *Archives of Pathology and Laboratory Medicine*, vol. 144, no. 2. College of American Pathologists, pp. 229–239, 2020. doi: 10.5858/arpa.2018-0477-RA.
- [22] C. H. G. L. C. V. E. G. Akrivi Katifori, "Ontology visualization methods—a survey," *AMC Journals*, 2007.
- [23] C. H. G. L. C. V. E. G. Akrivi Katifori, "Ontology visualization methods—a survey," *AMC Journals*, 2007.
- [24] E. Femi Aminu, I. O. Oyefolahan, M. Bashir Abdullahi, and M. T. Salaudeen, "A Review on Ontology Development Methodologies for Developing Ontological Knowledge Representation Systems for various Domains," *International Journal of Information Engineering and Electronic Business*, vol. 12, no. 2, pp. 28–39, Apr. 2020, doi: 10.5815/ijieeb.2020.02.05.
- [25] M. Uschold and M. King, "Towards a Methodology for Building Ontologies," 1995.
- [26] M. Fernández-López and A. Gómez-Pérez, "The integration of OntoClean in WebODE."
- [27] B. J. Wielinga, W. N. H. Jansweijer, G. Schreiber, B. Wielinga, and W. Jansweijer, "The KACTUS View on the 'O' Word," 2003.
- [28] H. S. Pinto, S. Staab, and C. Tempich, "DILIGENT: Towards a fine-grained methodology for Distributed, Loosely-controlled and evolving Engineering of ontologies." [Online]. Available: <http://www.ncbi.nlm.nih.gov/Taxonomy/>
- [29] G. A. V. & J. P. A. Konstantinos Kotis, "HCOME: A Tool-Supported Methodology for Engineering Living Ontologies," 2004.
- [30] D. M. Noy, "Ontology Development 101: A guide to Creating your First Ontology, Stanford Medical Informatics," 2000.

- [31] R. Navigli, A. de Nicola, and M. Missikoff, "A proposal for a unified process for ontology building: Upon Cite this paper A proposal for a Unified Process for ONtology building: UPON 1." [Online]. Available: <http://www.athena-ip.org>
- [32] T. Slimani, "Ontology Development: A Comparing Study on Tools, Languages and Formalisms," *Indian J Sci Technol*, vol. 8, no. 24, Sep. 2015, doi: 10.17485/ijst/2015/v8i1/54249.
- [33] S. Chowdhury and G. G. Chowdhury, "Using DDC to create a visual knowledge map as an aid to online information retrieval," Bates, 1987. [Online]. Available: <http://eprints.cdlibr.strath.ac.uk/2624/>
- [34] P. v. N. Rastogi, "Analyzing Ontology Editing Tools for Effective Semantic Information Retrieval," *International Journal of Engineering Sciences & Research Technology*, 2017.
- [35] Retrieved from : <http://www.mindswap.org/2004/>, "Swoop. Semantic Web Ontology Overview and Perusal," 2004.
- [36] T. Slimani, "Ontology Development: A Comparing Study on Tools, Languages and Formalisms," *Indian J Sci Technol*, vol. 8, no. 24, Sep. 2015, doi: 10.17485/ijst/2015/v8i1/54249.
- [37] M. S. Noman I, "A Dot NetBased Tool for Ontology Development and Editing," *ICCET*, 2010.
- [38] G. B. K. Č. R. L. J Bārzdīņš, "UML style graphical notation and editor for OWL 2," in *Conference on Business - Springer*, 2010.
- [39] N. Hossain, M. B. Sampa, F. Yokota, A. Fukuda, and A. Ahmed, "Factors Affecting Rural Patients' Primary Compliance with e-Prescription: A Developing Country Perspective," *Telemedicine and e-Health*, vol. 25, no. 5. Mary Ann Liebert Inc., pp. 391–398, May 01, 2019. doi: 10.1089/tmj.2018.0081.
- [40] S. Mohsin-Shaikh *et al.*, "The impact of electronic prescribing systems on healthcare professionals' working practices in the hospital setting: A systematic review and narrative synthesis," *BMC Health Serv Res*, vol. 19, no. 1, Oct. 2019, doi: 10.1186/s12913-019-4554-7.
- [41] P. Thesis and M. Poveda Villalón, "Ontology Evaluation: a pitfall-based approach to ontology diagnosis," 2016.

- [42] J. F. Ethier, F. Goyer, P. Fabry, and A. Barton, "The prescription of drug ontology 2.0 (PDRO): More than the sum of its parts," *Int J Environ Res Public Health*, vol. 18, no. 22, Nov. 2021, doi: 10.3390/ijerph182212025.
- [43] A. Grando, S. Farrish, C. Boyd, and A. Boxwala, "Ontological approach for safe and effective polypharmacy prescription." [Online]. Available: <http://www.nlm.nih.gov/research/umls/rxnorm/>
- [44] L. Puustjärvi, "The Challenges of Electronic Prescription Systems Based on Semantic Web Technologies."
- [45] S. H. Ghasemi *et al.*, "Design and evaluation of a smart medication recommendation system for the electronic prescription," in *Studies in Health Technology and Informatics*, 2019, vol. 260. doi: 10.3233/978-1-61499-971-3-128.
- [46] A. Khalili and B. Sedaghati, "Semantic medical prescriptions - Towards intelligent and interoperable medical prescriptions," in *Proceedings - 2013 IEEE 7th International Conference on Semantic Computing, ICSC 2013*, 2013, pp. 347–354. doi: 10.1109/ICSC.2013.66.
- [47] O.-S. Chirila, L. Stoicu-Tivadar, O.-S. Lupșe, and L. Stoicu-Tivadar, "Pediatrics Prescriptions with Ontologies and Treatment Suggestions. Computer Simulations for Predicting the Outcome of 3D Bioprinting View project Semantic Data Analysis in Medical Informatics View project Pediatrics prescriptions with ontologies and trataments suggests," 2014. [Online]. Available: <https://www.researchgate.net/publication/265095936>
- [48] O. S. Lupșe, C. B. Chirila, and L. Stoicu-Tivadar, "Harnessing Ontologies to Improve Prescription in Pediatric Medicine," in *Studies in Health Technology and Informatics*, 2018, vol. 255. doi: 10.3233/978-1-61499-921-8-97.
- [49] A. Khalili and B. Sedaghati, *Pharmer-Towards Semantic Medical Prescriptions*. 2014. [Online]. Available: www.slidewiki.eu
- [50] S. A. A. M. G Ghinea, "A Jini-based solution for electronic prescriptions," in *Ghinea, George, et al. "A Jini-based solution for electronic prescriptions." IEEE Transactions on Information Technology in Biomedicine*, 2006.

- [51] R. , S. E. , M. R. , R. R. , J. F. , & A. A. Fonseca-Pinto, "MOVIDA. eros: an eHealth Solution for Cardiac Rehabilitation Programs.," in *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO). IEEE*, 2020.
- [52] A. , M. P. , & K. D. Tagaris, "Implementation of a prescription fraud detection software using RDBMS tools and ATC coding," in *9th International Conference on Information Technology and Applications in Biomedicine (pp. 1-4). IEEE.*, 2009.
- [53] L. Puustjärvi, "The Challenges of Electronic Prescription Systems Based on Semantic Web Technologies."
- [54] O. S. Lupse and L. Stoicu-Tivadar, "Pediatrics Prescriptions with Ontologies and Treatment Suggestions," in *Studies in Health Technology and Informatics*, 2014, vol. 205. doi: 10.3233/978-1-61499-432-9-403.
- [55] C. G. Rodriguez-Gonzalez, A. Herranz-Alonso, V. Escudero-Vilaplana, M. A. Ais-Larigoitia, I. Iglesias-Peinado, and M. Sanjurjo-Saez, "Robotic dispensing improves patient safety, inventory management, and staff satisfaction in an outpatient hospital pharmacy," *J Eval Clin Pract*, vol. 25, no. 1, 2019, doi: 10.1111/jep.13014.
- [56] Pharm. D. , K. S. M. S. , L. C. C. M. B. S. Pharm. , S. A. S. Pharm. D. Connie A. Lin, "Drone delivery of medications: Review of the landscape and legal considerations," *American Journal of Health-System Pharmacy*, vol. 75, no. 3, pp. 153–158, Feb. 2018.
- [57] "Altexsoft.com drug interaction checker apis to minimize prescription errors," <http://https://www.altexsoft.com/blog/drug-interaction-checker-apis/>, accessed: Jul. 07, 2022.
- [58] and D. S. Hlomani Hlomani, "Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey," *IOS Press*, 2014.
- [59] AWS and Amazon, "Amazon Neptune - Fast, reliable graph database built for the cloud," *Amazon Web Services*. 2019.
- [60] M. Fernández, A. Gómez-Pérez, and N. Jurista, "METHONTOLOGY: From ontologica art towards ontological engineering workshop on Ontological Engineering," in *Spring Symposium Series*, 1997.

- [61] M. Fernández López, A. Gómez-Pérez, J. Pazos Sierra, and A. Pazos Sierra, "Building a Chemical Ontology Using Methontology and the Ontology Design Environment.," 1999. [Online]. Available: <http://www-ksl.stanford.edu:5915>
- [62] M. Ferndndez, A. G6mez-P~rez, and N. Juristo, "METHONTOLOGY: From Ontological Art Towards Ontological Engineering," 1997. [Online]. Available: www.aaai.org
- [63] M. A. Musen, "The protégé project," *AI Matters*, vol. 1, no. 4, pp. 4–12, Jun. 2015, doi: 10.1145/2757001.2757003.
- [64] M. Daquino, I. Heibi, S. Peroni, D. Shotton, and A. Haller, "Creating RESTful APIs over SPARQL endpoints using RAMOSE," *Semant Web*, vol. 13, no. 2, pp. 195–213, 2022, doi: 10.3233/SW-210439.
- [65] M. F. López, A. Gómez-Pérez, J. P. Sierra, and A. P. Sierra, "Building a chemical ontology using methontology and the ontology design environment," *IEEE Intelligent Systems and Their Applications*, vol. 14, no. 1, 1999, doi: 10.1109/5254.747904.
- [66] O. Corcho, M. Fernández-López, A. Gómez-Pérez, and A. López-Cima, "Building legal ontologies with METHONTOLOGY and WebODE," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2005, vol. 3369 LNAI. doi: 10.1007/978-3-540-32253-5_9.
- [67] T. Yogarajah, B. Kuhaneswaran, and S. Vasanthapriyan, "Evaluation and Validation of Subfertility Ontology and Decision Support System," in *5th SLAAI - International Conference on Artificial Intelligence and 17th Annual Sessions, SLAAI-ICAI 2021*, 2021. doi: 10.1109/SLAAI-ICAI54477.2021.9664683.
- [68] J. S. G. A. G. W. A. I. R. S. I. Wilson, "Ontology quality evaluation methodology," in *International Conference on Computational Science and Its Applications*, 2022.
- [69] Z. L. H. and D. Gao. Li, "Ontology construction and evaluation for chinese traditional culture: Towards digital humanity," *KO knowledge organization*, 2022.
- [70] M. Rospocher, S. Tonelli, L. Serafini, and E. Pianta, "Corpus-based terminological evaluation of ontologies," *Appl Ontol*, vol. 7, no. 4, pp. 429–448, 2012, doi: 10.3233/AO-2012-0114.

- [71] S. S. Tsaneva, "Human-centric ontology evaluation, Ph.D. thesis, Wien," 2021.
- [72] K. Benhocine, A. Hansali, L. Zemmouchi-Ghomari, and A. R. Ghomari, "Towards an automatic SPARQL query generation from ontology competency questions," *International Journal of Computers and Applications*, 2022, doi: 10.1080/1206212X.2022.2031722.
- [73] D. Wiśniewski, J. Potoniec, and A. Ławrynowicz, "BigCQ: A large-scale synthetic dataset of competency question patterns formalized into SPARQL-OWL query templates," May 2021, [Online]. Available: <http://arxiv.org/abs/2105.09574>
- [74] A. Gangemi, A. S. Lippolis, G. Lodi, and A. G. Nuzzolese, "Automatically Drafting Ontologies from Competency Questions with FrODO," Jun. 2022, doi: 10.3233/SSW220014.
- [75] J. F. Ethier, F. Goyer, P. Fabry, and A. Barton, "The prescription of drug ontology 2.0 (PDRO): More than the sum of its parts," *Int J Environ Res Public Health*, vol. 18, no. 22, Nov. 2021, doi: 10.3390/ijerph182212025.
- [76] A. Grando, S. Farrish, C. Boyd, and A. Boxwala, "Ontological approach for safe and effective polypharmacy prescription.," *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, vol. 2012, 2012.
- [77] L. Puustjärvi, "The Challenges of Electronic Prescription Systems Based on Semantic Web Technologies."
- [78] A. Khalili and B. Sedaghati, "Semantic Medical Prescriptions-Towards Intelligent and Interoperable Medical Prescriptions." [Online]. Available: <http://www.imi.europa.eu/>
- [79] A. Kilgarrieff *et al.*, "The sketch engine: Ten years on," *Lexicography*, vol. 1, no. 1, pp. 7–36, 2014, doi: 10.1007/s40607-014-0009-9.
- [80] "Team2-WordNet-module-instructor".