

Cloud-based Ontology Solution for Conceptualizing Human Needs



uOttawa

Shawn Dexter Dsouza

Faculty of Graduate and Postdoctoral Studies

University Of Ottawa

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science

© Shawn Dexter Dsouza, Ottawa, Canada, 2015

Acknowledgments

I'd like to take this opportunity to, first and foremost, extend my appreciation to my supervisor, Dr. Abdulmotaleb El Saddik, who has been a beacon of inspiration, and has set an example that I can only dream to follow in my career to come. He is a true and natural leader, who emboldened me to achieve a feat I never thought possible of myself. I hope to one day repay him for all the kindness, warmth and guidance he has granted me.

I'd also like to express my deep gratitude to Professor Liam Peyton who has always served as a pillar of wisdom and strength, not only through my grad-school, but also during my undergraduate years at this university. I'd also be remiss to not point out my gratefulness to Mukesh Saini who has made this entire experience an absolute joy while also guiding me in the right direction. He has shown me what it truly means to do everything in life with a smile on your face.

Finally, I'd like to thank and dedicate this thesis to my family in Bombay, who have been through some extremely rough years in the past decade, but have unceasingly stood by me and my dreams to the very end. To Miqdad Jaffer, who has been my mentor, a friend and a brother for the past five years, without whom I'd have stumbled all over myself countless of times. And to Bryan Le, who has been family and more to me ever since I got to Ottawa, and has proven to always be by my side as I traverse through this life. I am truly blessed to have him in my life.

None of this would have been possible without the help from the people I have mentioned here. I thank you all.

Abstract

The current generation has seen technology penetrate every aspect of our life. However, even with recent advancements, adopters of contemporary technology are often angry and frustrated with their devices. With the increasing number of devices available to us in our day-to-day lives, and with the emergence of newer technologies like the Internet of Things, there is a stronger need than ever for computers to better understand human needs. However, there is still no machine understandable vocabulary that conceptualizes and describes the human-needs domain. As such, in this thesis we present a cloud-based ontology solution that conceptualizes the needs-domain by describing the relationships between the concepts of an Agent, a Role, a Need, and a Satisfier.

The thesis focusses on the design of an OWL ontology which is based on an existing human-needs model. The human-needs model chosen for the ontology stems from a trans-disciplinary approach led by Manfred Max-Neef, called the Fundamental Human Needs model. It is seen as classifiable, finite and constant across all cultures and time periods. The methodology approach used to develop a new ontology is *METHONTOLOGY*, which is geared toward conceptualizing an ontology from scratch with the mindset of continual evaluation.

We then further discuss the overall FHN Ontology comprising of various components including a RESTful Web Service and a SPARQL endpoint for querying and updating the FHN Ontology. The ontology is evaluated via competency questions for validation and via the Ontology Pitfall Scanner for verification and correctness across multiple criteria. The entire system is tested and evaluated by implementing a native android application which serves as a REST client to connect to the FHN Ontology end-point.

Table Of Contents

Acknowledgments	ii
Abstract	iii
Table Of Contents	iv
Table Of Figures	vii
List Of Tables & Charts.....	viii
List of Code-Snippets	viii
List of Screen Shots.....	viii
Ch 1. Introduction	1
1.1. Motivation	2
1.2. Problem Statement.....	4
1.3. Proposed Solution and Contribution	4
1.4. Thesis Methodology, Organization & Outline	5
Ch 2. Background.....	8
2.1. Related Works	8
2.1.1. Identifying User Needs through Social Media	9
2.1.2. OLA: Ontology for Predicting Students Emotion's.....	10
2.1.3. SmartMuseum: Personalized Context-aware Recommender System.....	12
2.1.4. MCRS: The Hybrid- ϵ -greedy Algorithm	13
2.1.5. Comparison of Related Works to FHN Ontology	15
2.1.6. The 'Linking Open Data' Project.....	16
2.2. Literature Review.....	19
2.2.1. Consumer Behavior & Maslow's Universal Hierarchy of Needs	19
2.2.2. Max Neef's Fundamental Human Needs	20
2.2.3. Max Neef's FHN vs Maslow's Hierarchy.....	21
2.2.4. Distinguishing between Needs and Satisfiers	21
2.3. OWL & Ontologies	24
2.3.1. Exploring Ontologies	24

2.3.2.	The OWL Language.....	27
2.3.3.	Definitions & Key Terms.....	30
2.3.4.	Ontology & Semantic Web Tools	31
Ch 3.	Engineering the FHN Ontology	33
3.1.	Knowledge Base Systems Vs Ontologies	34
3.2.	Methontology.....	35
3.3.	Competency Questions.....	36
3.4.	Specification.....	40
3.5.	Conceptualization Of The FHN Ontology	41
3.5.1.	Entities.....	43
3.5.2.	Integration of CoDePs	55
3.5.3.	Properties	57
3.5.4.	SWRL Rules	60
3.6.	Implementation	64
3.6.1.	The FHN Technology Stack.....	68
Ch 4.	FHN Ontology System	73
4.1.	Cloud Based System: Software As A Service.....	74
4.2.	System Components	74
4.3.	The FHN Domain Model.....	77
4.4.	Application Sequence Interaction	81
4.3.1.	Sequence Interaction	81
4.5.	Testing the System.....	83
4.4.1.	FHNReco: Native Android App	84
Ch 5.	Evaluation	88
5.1.	Background on Ontology Evaluation	88
5.2.	OOPS! Ontology Pitfall Scanner	89
5.3.	Evaluating Verification : Correctness & Completeness	92
5.3.1.	Structural Dimension Evaluation.....	93
5.3.2.	Functional Dimension Evaluation.....	94

5.3.3.	Usability Profiling Dimension	95
5.3.4.	Consistency Criteria.....	96
5.3.5.	Conciseness Criteria	97
5.4.	Evaluating via Competency Question	98
5.4.1.	Setting Up Individuals and Asserted Facts	99
5.4.2.	Analysis of Inferred Facts on Individuals.....	101
5.4.3.	Cross Referencing Inferred Facts to CQs.....	104
5.5.	Discussion on Evaluation	105
Ch 6.	Conclusion	107
6.1.	Thesis Contribution.....	108
6.2.	Concluding Statements & Experiment Results	109
6.3.	Future Work.....	110
References		112

Table Of Figures

Figure 2.1: OLA Ontology Segment illustrating the OCC Emotion Variables.....	11
Figure 2.2: Illustration of the Mobile Context Aware Recommender System.....	15
Figure 2.3: Portion of the Linking Open Data Cloud Diagram [93].....	18
Figure 2.4: Conceptual difference between ‘Needs’ & ‘Satisfiers’	22
Figure 2.5 OWL Sub Languages.....	29
Figure 3.1: Competency Question Documentation for the FHN Ontology.....	38
Figure 3.2: Specification Document for the FHN Ontology.....	38
Figure 3.3: Conceptual Model of the FHN Ontology depicting major entities.....	42
Figure 3.3: Conceptual model between Need and Satisfier.	44
Figure 3.4: Relationship between entities Satisfier and Existential Category.....	47
Figure 3.5: The Existential Category entity along with its associated children.....	47
Figure 3.6: Conceptual Model showing the Need, Satisfier and Role.....	50
Figure 3.7: Conceptual model between Role , Need and Agent.....	52
Figure 3.8: Conceptual model between the Destroyer and Satisfier.....	54
Figure 3.9: CoDeP within FHN Ontology.....	56
Figure 3.10: SWRL Rules currently in the FHN Ontology.....	63
Figure 3.11: Screenshot of entity-structure of the FHN Ontology.....	65
Figure 3.12: FHN Ontology as visualized by the Protégé tool.....	67
Figure 3.13: FHN Technology Stack based on the Semantic Web Stack.....	69
Figure 4.1: Component Diagram of the FHN Ontology System.....	75
Figure 4.2: UML Diagram of the FHN Domain Mode.....	78
Figure 4.3: Sequence Diagram depicting flow of events from Client to Server.....	83
Figure 5.1: Evaluation structure of pitfalls for classification via dimensions.....	93
Figure 5.2: Evaluation structure of pitfalls for classification by criteria.....	93
Figure 5.3: CQ Document.....	101

List Of Tables & Charts

Chart I: Evaluation classified via Structural Dimension between FHN Ontology and others ..	93
Chart II: Evaluation classified via Functional Dimension	94
Chart III: FHN Ontology evaluation comparison by Usability Dimension.....	95
Chart IV: Criteria Evaluation comparison by Consistency	97
Chart V: Criteria Evaluation comparison by Conciseness.....	98

List of Code-Snippets

Code Snippet 1: OWL/RDF for Satisfier entity in the FHN Ontology.....	44
Code Snippet 2: OWL/RDF for ExistentialCategory entity	46
Code Snippet 3: OWL/RDF for the Need entity in the FHN Ontology	49
Code Snippet 4: OWL/RDF of Role entity with restrictions entity	51
Code Snippet 5: RDF/OWL for Destroyer entity with restrictions	54
Code Snippet 6: Relationship property “ <i>isASatisfierFor</i> ” in the FHN Ontology	58
Code Snippet 7: Maven/Gradle dependency for Android App	84

List of Screen Shots

Screenshot I: FHN Android App using Google+ Profile Info	86
Screenshot II: Android App displaying query results from FHNRest Service & DBPedia	87
Screenshot III: FHN Ontology Individuals for evaluation in Protégé.....	100
Screenshot IV: Inferred Facts (yellow highlight) and asserted facts for Agent02.....	103
Screenshot V: Inferred Facts(yellow highlight) with no assertions for Sat06	104

List of Abbreviations

CQ: Competency Questions

CODEP: Content Ontology Design Pattern

FOAF: Friend Of A Friend Ontology

FHN: Fundamental Human Needs Ontology

IoT: Internet of Things

KBS: Knowledge Based Systems

KEs: Knowledge Engineers

OOPS: Ontology Pitfall Scanner.

OWL: Web Ontology Language

POJO: Plain Old Java Object

RDF: Resource Description Framework

SWRL: Semantic Web Rule Language

SPARQL: SPARQL Protocol and RDF Query Language (recursive acronym)

Introduction

In the past few years the Semantic Web has been picking up pace with surged acceleration. There has been increased awareness drawn toward the need for an extended web which will allow for machine-to-machine understanding. The Semantic Web allows for a standardized infrastructure for databases, services, programs, sensors and personal devices to exchange and share data within a semantic context [1]. Thus, contrary to popular belief, the Semantic Web will not only provide for an infrastructure for the web but also other cutting edge technologies. These aforementioned elements are the focus within emerging technologies such as the Internet of Things (IoT) and Smart Cities. Furthermore, semantic technologies have played a significant role in improving technologies pertaining to recommender systems by working synergistically with them to avoid issues like “interest-acquisition and cold-starts [40].

Recommender systems, themselves, are gaining popularity amongst consumers, and are garnering elevated attention from research and development. A competition held by an internet media-streaming giant, Netflix, awarded US\$1,000,000 for anyone or team that improve Netflix’s recommender system by at least 10.06% [98]. In an effort to improve their offerings, developers of recommender systems have incorporate ontologies into their technology stack [43]. However, the developers of these systems use the existing ontologies that are available to them since ontology design and development is a feat in itself.

As such, in an effort to further supplement the power of emerging technologies as well as that of recommender systems, we design and implement a cloud based ontology system that conceptualizes the fundamental human needs which serves as a common vocabulary for data integration and knowledge inference. The ontology is accessible through a SPARQL endpoint which points to a repository that holds the FHN Ontology. Furthermore, we develop a RESTful web-service that allows for querying the FHN Repository, which in turn queries external linked-data projects such as DBPedia. We test this system via a native-android application that serves as a REST-client. The ontology is evaluated for correctness and completeness using an ontology anomaly and design pitfall detection tool, and also CQ-driven approach in order to verify validation.

1.1. Motivation

Technological advancements have empowered humans to allow themselves an increased quality of life in a variety of ways. However, even adopters of contemporary technology are often angry and frustrated with their devices. This frustration stems from the fact that the computers we use are not in harmony with our needs. In his book, *Leonardo's Laptop: Human Needs and New Computing Technologies*, Ben Shneiderman makes a noteworthy distinction between “old computing” and “new computing”. He states “computing today is about what computers can do; the new computing will be about what people can do” [84]. The book talks about how computer systems need to be in better sync with human needs.

Another contributing factor toward the frustration faced by consumers stems from *decision fatigue* – a phenomenon which occurs when an individual is faced with a series of decisions over a period of time [85][86]. With the increasing number of devices in our day-to-

day lives, along with several options in other areas like shopping for groceries and clothes; people are known to make progressively less favorable decisions as the day passes. The increase in number of devices leads to even more choices and decisions that need to be made. While, these devices may allow for more functionality, one could argue that the sheer number of devices lead to decreased productivity, especially when these devices don't truly understand our needs.

It is estimated that the internet will connect 25 billion devices by 2015 and around 50 billion devices by 2020 [4]. One of the primary goals of the interconnecting of devices is to create situation awareness and allow for humans and machines to have a better understanding of their context. In the current world of the web, computers, machines and other devices have a very limited or even no understanding of the information they relay to each other as well as the information relayed to humans. With a common and shared vocabulary, machines can use the data available on the web to further understand the needs of humans. The volume of data on the Web has been growing at a stunning rate. It is estimated that everyday around 2.5 quintillion bytes of data is created and that 90% of data today was only generated in the past two years [30].

This thesis, hence, aims toward providing a shared and interoperable vocabulary that allows for computer systems to better understand and conceptualize human needs in an attempt to eventually provide trimmed, uncluttered recommendations and hence reduce decision fatigue.

1.2. Problem Statement

While there are several ontologies that target various domains, there seems to be no effort directed toward conceptualizing a vocabulary that encapsulates the human needs domain. Recommender systems target various domains, and use ontologies whenever available and possible. However, without a vocabulary that conceptualizes the fundamental needs of a human being, the task will prove to be limited; especially when the one common domain of every recommender system is to serve a human need.

Researchers & developers of recommender systems and similar technologies have begun to use ontologies to represent complex and relationship-rich data. An example of this is the Mobile Context Aware Recommender System developed by researchers who use various *existing* ontologies to capture complex data. However, the lack of an ontology available to conceptualize the “needs” domain limits the system severely. A vocabulary that conceptualizes the relationship between a human, his various roles and the associated needs to these roles will allow a recommender system to make more meaningful recommendations by understanding more about the needs of the agents.

Hence, there is an inherent need for such a knowledge representation that is machine readable as well as human-readable. Furthermore, with growing and dynamic data that will be accessible, reasoning and inference abilities should also be available.

1.3. Proposed Solution and Contribution

The previous discussion outlined the need for a semantic approach for conceptualizing fundamental human needs. The inherent need for a machine and human readable vocabulary with reasoning and inference abilities is perfectly suited for system using the Web Ontology

Language (OWL)

As such, this thesis presents a cloud based scalable ontology system that conceptualizes the fundamental needs of humans. It outlines the iterative conceptualization and development of an OWL Ontology, namely the FHN Ontology that decouples the concepts of *agents*, *roles*, *needs* and *satisfiers*. Such a conceptualization allows for flexibility where agents can have multiple *roles*, and roles can have multiple *needs*. The ontology can thus be scaled by importing external *roles* that conform to the FHN Ontology. The system provides a SPARQL endpoint that can be linked to other ontologies in the semantic web, and thus allowing for contribution to the Linking Open Data. A user may connect to the endpoint which will give him access to a repository that will allow for querying, writing and updating the ontology with additional data when needed.

1.4. Thesis Methodology, Organization & Outline

In 2004 Daniel Fallman proposed the distinction between Design-oriented research and research-oriented design in the field of Human Computer Interactions. He stated that design-oriented research entails the revealing of a truth or new knowledge which was only attainable via the design phase of the research. Hence the “artifact” is not the design, but the new knowledge that was attained. In contrast, in research-oriented design the “artifact” is the design itself and the efforts put into throughout the research phase [34].

Even though the main aim of the Fundamental Human Needs Ontology is to be able to deduce new knowledge (based on current available knowledge), the focus of this thesis was directed toward the designing such an ontology that would allow for that.

Research in the area of the Internet of Things and Web Semantics are only just begun

to pick up pace. At the current state, there are very few Ontology-based applications and applications that use Web Semantics as their foundation. Since the research focus of this thesis is still in its early stages the methodology used is research oriented design [33].

However, ontology-design in itself is has no accepted standard. Therefore, research on the methodology for designing the underlying ontology for the system had to be performed. After examining various methodologies such as the *NeOn Methodology* [87], *Model Driven Ontology* [88] and the *UPON Methodology* [89]; we decided that the design approach best suited for our work would be one proposed by Gomez-Perez, namely *METHONTOLOGY* [57].

In this thesis we perform a gap analysis of the approaches proposed in existing literature and propose an ontology based system that addresses the problem statement along with the gaps. The thesis first highlights the conceptualization process of the FHN Ontology then moves onto describing, with illustrations, the entire FHN Ontology System which encompasses the application layer, the RESTful web-service as well as the FHN Ontology repository.

Evaluation of the design is then performed based on a set of criteria known as Competency Questions (CQs) Below are the steps followed in this thesis methodology:

1. Gap analysis and problem identification based on existing literature
 - a. Comparative study between various human needs models and identification methodologies in psychology, sociology, economy etc.
 - b. In-depth study of current ontology design and comparison between well-defined ontologies to target suitable design principles used.
2. Conceptualization of Fundamental Human Needs Ontology Model
 - a. Conceptualization of entities: class representation, sub-classes & individuals

- b. Iterative approach to developing properties expressing relationships, and defining appropriate domain and ranges.
 - c. Formulation of the Semantic Web Rule layer that extends the ontology
- 3. Development and implementation of the application layer
 - a. Description of the various components within the system
 - i. The Domain Model
 - ii. The FHN RESTful Web Service
 - iii. The FHN Sesame Repository + FHN Ontology
- 4. Evaluation of the Ontology
 - a. Evaluation based on various ontology evaluation tools used available
 - b. Evaluation of the Ontology based on Competency Questions
- 5. Conclusions and future work.

The thesis organization is as follows. Chapter 2 provides background information according to step 1 of our methodology. Chapter 3, 4 and 5 implement step 2, 3 and 4 of the ontology design process, i.e., ontology conceptualization, application layer design, and evaluation. Finally, we provide conclusions and future work in Chapter 6.

CHAPTER 2

Background

In the previous section we discussed the motivation, stated the problem statement and proposed our solution to the problem. Before moving on to the conceptualization and design of the actual ontology and its system, it's imperative that we discuss the background research, literature and overall composition of OWL ontologies.

Hence, in this section we briefly discuss some of the more important related works and efforts in similar areas of research. Furthermore, we discuss the literature surrounding human needs - more specifically Max Neef's Fundamental Human Needs and how it compares to Maslow's Hierarchy. Finally, we discuss OWL and Ontologies, which sets us up for the next chapter where we discuss the actual engineering of the FHN Ontology

2.1. Related Works

There has been little to no effort directed toward building a semantic application that would serve as knowledge representation and identification of human needs. However, there have been efforts to identify Human Needs in other areas and interests of research. In this section discuss the recent effort to target the human needs domain along with other efforts

directed toward ontology engineering, ontology inference systems on human behavior as well as the use of OWL ontologies in recommender systems.

2.1.1. Identifying User Needs through Social Media

Recent efforts have been made toward utilizing social media and linguistic analysis to identify individual user needs [1]. In a paper headed by the IBM Research division, the authors developed a psychometric model for universal needs via a crowd sourced study [7]. Furthermore, the authors explore the linguistic relationship between people's needs and their social media data. They evaluate the effectiveness of their needs model by examining the purchasing behavior of social media users. They then examine the consistency between the purchasing behavior predicted by their needs model and the needs that would be prescribed by the universal needs. Their results suggest that their needs modeling approach is effective in predicting *purchasing behavior*.

Their efforts are geared toward developing an instrument for fundamental human needs. However the underlying needs theory used to model the data uses Maslow's hierarchy. Maslow's model has already been critiqued to be lacking [11] [12] [13]. There have been other works that point toward how Maslow's theories have been used to make people "want to consume" instead of helping satisfy their needs to facilitate their development. These efforts toward identifying user needs through social media were geared toward a more marketing and consumer perspective and don't fully address the fundamental universal needs that are required to address the situation fully [12] [14].

The study was directed toward identifying needs through data content on the web. However the focus of these results has been to predict the purchasing behavior of humans

rather than providing solutions to various “needs” that a human may need “satisfied”. Furthermore, the need model used, namely Maslow’s Hierarchy of Needs, has cultural and hierarchical limitations that may not fit across borders. The “needs model” used in this thesis is Manfred Max-Neef’s Fundamental Human Needs model that was modeled to be constant through all human cultures as well as through historical time periods [8].

2.1.2. OLA: Ontology for Predicting Students Emotion’s

The OLA ontology aims to predict the emotions of a student during an exam or a test. Since e-learning systems are used by a wide variety of students with different skills, backgrounds and preferences, there was need for personalization of the learning context. Emotions were considered to be a key issue in such tasks [35]. The OLA ontology uses the OCC psychological model as its base model. The OCC model is based on cognitive approach of emotions that explains the origins of emotions by describing the cognitive process that instigates them [36]. By using the classification of emotions as described in the OCC model, these computer science researchers were able to build an emotion predicting system by using the inference capabilities of OWL. They were able to do this by using a set of rules and axioms.

An example for an axiom, which predicts the emotional state of Satisfaction, is as follows:

$$\forall e \ (hasCause(e, c) \wedge hasSubjectOfEmotion(e, m) \wedge hasOccurrence(e, a) \wedge hasAppraisal(e, d) \rightarrow is_aSatisfaction(e))$$

The axiom above simply states that if an emotion that is caused for a reason and the subject of the emotion is *myself*, and if the occurrence of the emotion is *actual* and if the *appraisal* is *desirable*, then the emotion is inferred as *Satisfaction*. Since the OLA ontology is

still in its developmental stage, it currently has a prediction rate of around 47%.

However, the researchers note that some were more easily predicted by the ontology than others. For example, the “Disappointment” was predicted 80% of the time while “Distress” was predicted only 43% of the time [37]. **Figure 2.1** illustrates the OLA ontology based on the OCC emotion variables. This ontology depiction includes the concepts along with the associated individuals/instances of the ontology.

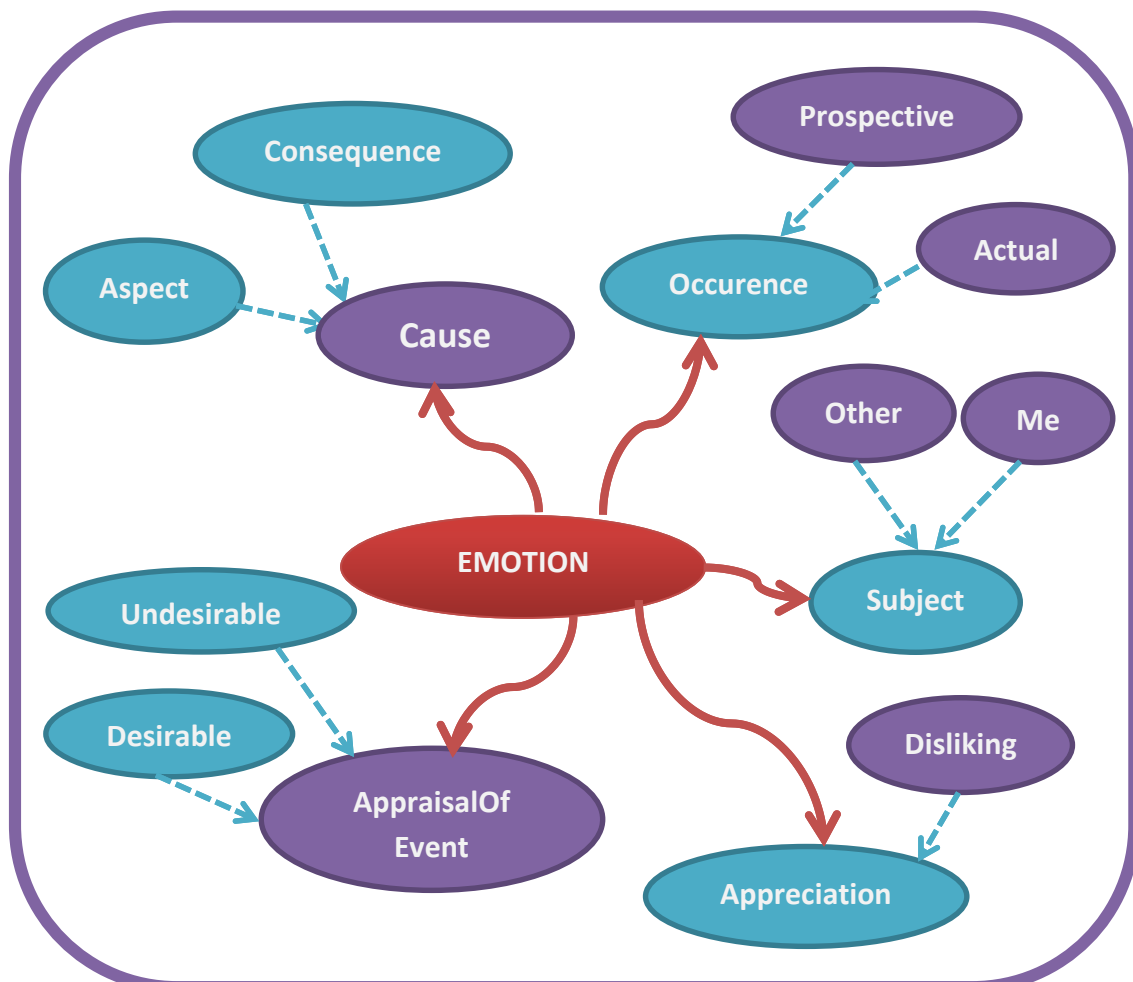


Figure 2.1: OLA Ontology Segment illustrating the OCC Emotion Variables

2.1.3. SmartMuseum: Personalized Context-aware Recommender System

In the digital cultural heritage collections space, there is need for interlinking of data of the mutually interrelated data that are provided for users through these digital collections and libraries [38]. Semantic web technologies have allowed for this interlinking of data [39]. With the rise of popularity of ubiquitous computing, there are an increasing number of users accessing web-based digital services while on the go through mobile devices. This opens up an opportunity, as well as stronger need for recommender systems to be able to assist users to find contents that are of interest. However, traditional recommender systems had pitfalls and suffer from a variety of problems. Some of them are listed as follows:

- i. **Cold Start:** A recommender systems thrive on data. However, when a user is using the system for the first time, the recommender system has very little context and user profile data. Hence the users cannot immediately benefit from such a system.
- ii. **Sparsity:** When a recommendation system is working on heterogeneous data (such as heritage collections), there are severe limitations pertaining to interlinking. This can lead to limited and poor recommendations. There is an inherent need for semantic technologies with interlinking capabilities to reduce sparsity.
- iii. **Over Specialization:** Closely related to the issue of sparsity that recommender systems face, recommender systems also suffer from over specialization of recommended data. With limited features of the items available to be recommended due to lack of interlinking of data across libraries, recommendations are sometimes too similar to the previous recommendation.

The SmartMuseum Recommender system uses semantic technology (ontologies), background knowledge, semantic reasoning via axioms and context aware user profiling to overcome the problems listed above [40].

2.1.4. Mobile Context-Aware Recommender Systems: The Hybrid- ϵ -greedy Algorithm

Djallel Bouneffouf et al. [42] describe their ongoing work on the implementation of a Mobile Context Aware Recommender System (MCRS) based on a hybrid- ϵ -greedy algorithm. The algorithm combines content-based filtering and case based reasoning techniques with the standard ϵ -greedy algorithm. Context aware systems as well as recommender systems are both used independently to facilitate users with more relatable information and service. While recommender systems are based on user interests, context-aware systems are based on the user's context. The MCRS is an effort toward combining user interests along with context awareness as the next logical step [42].

Thorough research efforts have been put toward recommender systems. Previous techniques in developing recommender systems were based mainly on computational behavior of the user in order to model his interests [43] [44]. The context variables of the user were not taken into consideration. Context variables include location, time and nearby people and events. The researchers aim to tackle to key issues with their MCRS system. The first aim is to develop a recommendation technique that follows the evolution of the user interests, and the second aim is to model the user's context in relation to the user's interests.

They develop a user model that will facilitate the system with capturing of context via mobile capabilities such as GPS or by inferring users location based on user's calendar or other devices

that they are connected to in the neighborhood. Furthermore, the user model facilitates reasoning and learning capabilities to be able to deduce high level information such as identifying the user's situation. The proposed user model diagram for the Mobile Context-Aware Recommender System is illustrated in the **Figure 2.2.**

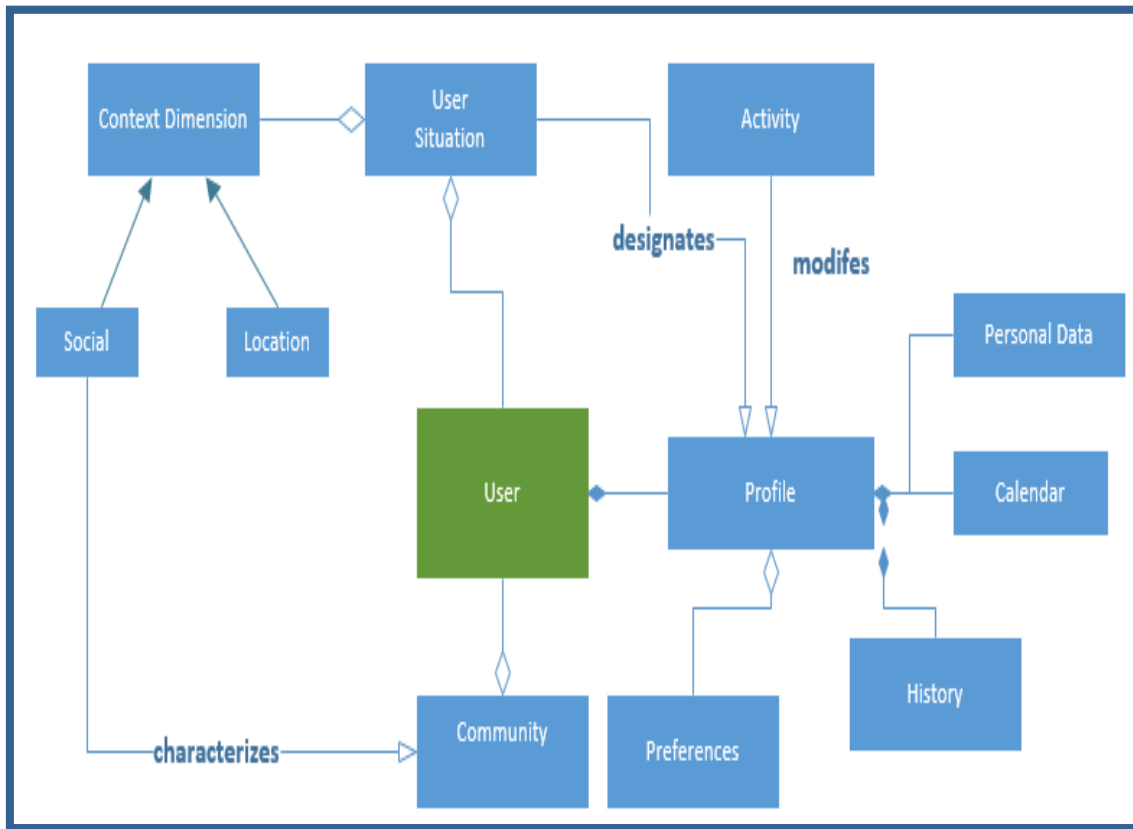


Figure 2.2: Illustration of the Mobile Context Aware Recommender System

The model is composed of various data extracted from the user's profile as well as contextual or situational information of the user. This includes dynamic information as well as

user inputted preferences. Profile information consist of user history, user preferences etc. Situational and contextual information consists of the user's location, time, and user's social connections.

Representing information like time and social context can be complex. Temporal information is continues and can be depicted at several layers of granularity. To allow for a well-defined representation of the temporal information and its manipulation, the researchers use the OWL-Time Ontology.

On the other front, representing social connections entails identifying information such as the user's close friends, important business colleague or customer or his manager/employee. Once again a clear and well-defined model for representation and reasoning on social clustering was necessary. To achieve this Djallel Bouneffouf et al. [42] use the FOAF OWL-Ontology to describe the social network between a set of concepts and properties.

2.1.5. Comparison of Related Works to FHN Ontology

In this subsection, we do a comparison of the related works we discussed in the previous subsections. The table below compares the various works in chronological order while highlighting the major purpose of each work. The related works share a similar theme. One one hand we have Y. L. Huahai Yang from the IBM Research division who attempts to identify human needs through social media, while on the other hand we have two recommender systems by D. Bouneffouf et al. and T. Ruotsalo et al. respectively who use ontologies to improve their systems in order to better serve human needs. Finally we have the OLA Ontology who creates an ontology to conceptualize human emotions. As we can see,

there have been recent efforts that touch on similar interests, but as of yet no one has attempted to conceptualize the human needs domain through an ontology vocabulary.

Name	Year	Purpose
Identifying User Needs	2013	Capturing Human Needs through Social Media
MCRS	2012	Mobile Recommender system using ontologies
OLA Ontology	2011	Ontology to predict human emotions
SmartMuseum	2008	Context-aware recommender using ontologies
FHN Ontology	2015	Ontology to conceptualize and predict human needs

2.1.6. The ‘Linking Open Data’ Project

One of major selling points of the semantic web is that it can perform day-to-day investigations for human-beings [6]. One of the major reasons this is possible is because of the concept of linked data, upon which the Semantic Web puts heavy emphasis. In another paper, Tim Berners Lee stated that the Semantic web isn’t just about putting meaningful data on the internet, but it’s about making links between this data to allow of interoperability and exploration of this data by humans and machines alike [47].

The concept of Linked Data is about connecting data between different data sources by employing the RDF (Resource Description Framework) and HTTP protocol to publish structured data on the web. This will allow for data in one data source to be linked to data in a different data source [46]. There are similarities between the resulting Web of Data and today’s Web of documents. The current webs of documents allow users to use HTML browsers to follow documents from one to another using HTML links. Similarly, Linked Data browsers

allow for users to navigate between different data sets using RDF links. A Linked Data browser will facilitate a user with the ability to move through a potentially endless Web of data sources that are interconnected by RDF links. Christian Bizer et al. [46] said that hypertext links between HTML pages plays the role of the glue that holds together the Web of Documents, while the RDF links are the glue that holds together the Web of Data.

‘Linking Open Data’ is a project that was founded in 2007 by a grassroots community, and is strong evidence of the emergence of the Web of Data, which at one point was only a vision [93] [95]. The aim of the project was to identify available open license data sources and to republish them in RDF format on the Web. These data sources would then be interlinked with each other.

The size of the Web of Data grew to over two billion RDF triples during the year 2007. This spurt of growth is attributed directly to the efforts of the community behind the Linking Open Data project. The triples originate from various domains such as geographic information, census information, people etc. Some of the key interlinking data sources are DBpedia [94] , that extracts triples from “Infoboxes” that are seen on the right hand side of Wikipedia articles, and Geonames that provides information on millions of geographical locations worldwide [96] [6]. **Figure 2.3** illustrates a portion of the Linking Open Data graph cloud model.



Figure 2.3: Portion of the Linking Open Data Cloud Diagram [93]

2.2. Literature Review

Many of us are familiar or have heard of Maslow's Hierarchy of Needs. Maslow's model is widely accepted and used in consumerism based marketing and also in motivational based management. However, while Maslow's model is widely accepted and has been proven to be effective, there are some fundamental flaws that the model holds. In the following subsections we briefly discuss Maslow's model, its flaws and then talk about why we picked Max-Neef's Fundamental Human Needs model as the base of our ontology. We also discuss one of the core aspects of Max-Neef's model - the difference between *needs* and *satisfiers*, which serves as a cornerstone of the FHN Ontology.

2.2.1. Consumer Behavior & Maslow's Universal Hierarchy of Needs

In 1943 Abraham Maslow theorized the patterns of Human Motivation which paralleled various other theories from Human Developmental Psychology [9]. Maslow's system had five levels of needs: physiological, safety, love, belonging, esteem and self-actualization. One of the core concepts was that the lower level needs have to be reasonably satisfied before the higher needs were sought after. There was, therefore, an explicit hierarchy of needs established [10]. However the very hierarchy and core concept that was established in Maslow's theory has been criticized and its effectiveness has been questioned [11]. Furthermore Maslow's explanation on human development has been said to be highly misstought and essentially reductionist [12].

While Maslow's theory attempts to capture the needs of a human in an individualistic society, criticism is found in the fact that he fails to address the cultural differences that would stem from the needs in collectivist society. Needs in a collectivist society would place priority

on community acceptance rather than self-centered needs [13]. These issues and holes in Maslow's theory raise several other questions. Some of the major questions being: Does one layer in the hierarchy really need to be satisfied before attempting to satisfy the next? Is a hierarchy even necessary?

2.2.2. Max Neef's Fundamental Human Needs

In his book, Manfred Max-Neef proposes an orientation that allows for a new set of practices based on Human Scale Development [8]. This development is directed toward the satisfaction of the fundamental human needs along with the facilitation of increased autonomy and self-reliance.

The Human Scale Development comes together with a transdisciplinary approach with major contributions coming from economy, sociology and philosophy; while additional contributions all the way from psychiatry & political science to engineering and law. This approach allows for a greater understanding by reaching beyond the limitations of strict disciplines. A concept that is as personal as "satisfiers to human needs" would greatly benefit from an approach that allows better understanding by surpassing disciplinary frontiers and allowing for an intersection between fields.

Max-Neef addresses and examines core questions such as what would be the best development process that will allow for greatest improvement in one's quality of life. This, in turn, leads to a follow the follow up question: What exactly determines quality of life? Human Scale Development states that quality of life depends of the possibilities people have to adequately satisfy their fundamental human needs. This finally leads to questioning what these fundamental human needs are.

2.2.3. Max Neef's FHN vs Maslow's Hierarchy

In the previous section we discussed Maslow's Hierarchy of needs along with its major shortcomings and criticisms received. Some of the points discussed were its questionable need of a hierarchy, its approach toward human development and how Maslow's needs are geared toward a more individualistic society with self-centered needs [11][12][13].

While Maslow's Hierarchy has proven to be useful, especially in the consumer and marketing universe, its shortcomings are very apparent. Manfred Max Neef's Fundamental Human Needs seems to target the very same shortcomings that Maslow's Hierarchy displays. The notion perpetuated by Maslow's Hierarchy has been used in consumerism and marketing theory to escalated poverty by promoting increased consumptive patterns.

Manfred Max-Neef states that "development is about people, not about objects". He proposes an alternative view and rejects the notion of a need for a "hierarchy". Instead, Max-Neef focusses on a constellation of fundamental and universal needs that are integrative and additive. The Fundamental Human Needs model attempts to fulfill needs in order to leave the human "satisfied" instead of needing to consume more to feel fulfilled. Furthermore, unlike Maslow's Max-Neef's FHN is consistent across individualistic societies with self-centered needs as well as collectivistic societies who place the needs of the community before theirs.

2.2.4. Distinguishing between Needs and Satisfiers

Manfred argues that contrary to traditional belief human needs are finite, consistent through all time, culture and historical period. It is suggested that the claim that human needs are infinite and change across culture and historical period is inaccurate and stems from misunderstanding of the concept of Needs and Satisfiers. Distinguishing between the two

concepts is crucial for epistemological and methodological reasons.

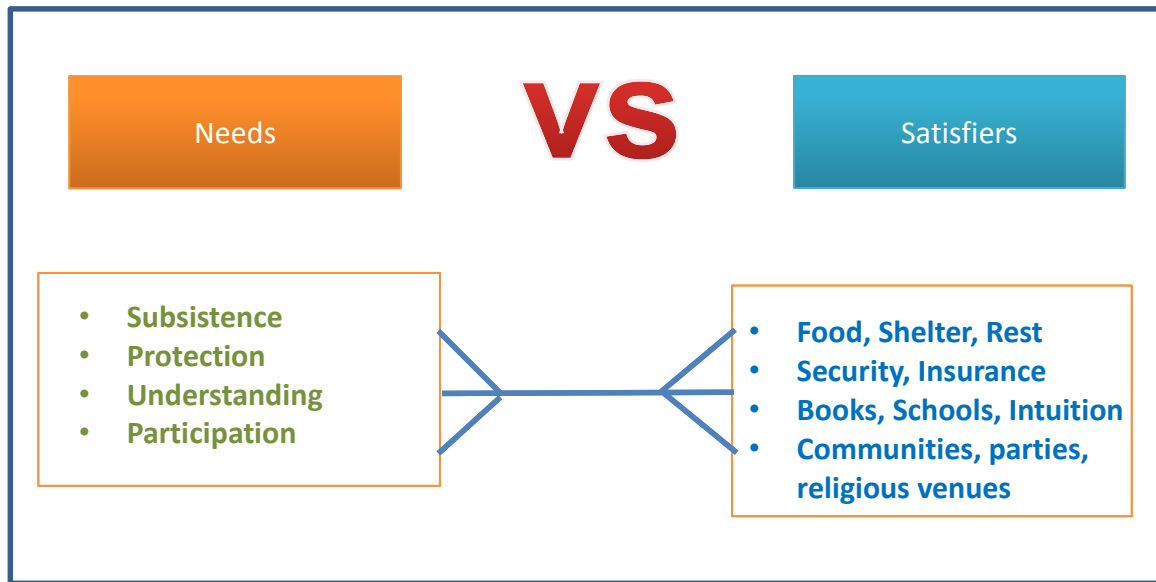


Figure 2.4: Conceptual difference between 'Needs' & 'Satisfiers'

Human Scale Development organizes human needs according to two categories based on the premise that human needs can be satisfied according to many criteria. These two categories are “Existential” and “Axiological” needs. The existential needs are “Being”, “Having”, “Doing” and “Interacting” while the axiological needs are “Subsistence” , “Protection” , “Affection” , “Understanding” , “Participation” , “Idleness”, “Creation”, “Identity” and “Freedom”.

Hence, with this classification being finite and fixed through all cultures and times proposes that the axiological and existential needs are the fundamental human needs of human beings and humans find satisfiers to fulfill these needs. For example, instead of viewing “food” and “shelter” as needs for a human being, we instead consider them satisfiers to the need for Subsistence.

Figure 2.4 depicts the conceptual differentiation between Needs and Satisfiers. The table below illustrates the matrix that outlines the existential and axiological needs relationship along with where the satisfiers fit into the picture.

Existential	Being	Having	Doing	Inter
Axiological				
Subsistence	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>
Protection	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>
Affection	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>
Understanding	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>
Participation	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>
Idleness	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>
Creation	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>
Identity	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>
Freedom	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>	<i>(list of satisfiers)</i>

Table 1: The Fundamental Human Needs Model

2.3. OWL & Ontologies

The previous section outlined the major concepts, aim and various technologies in the Semantic Web. The section also illustrated exactly where ontologies fit into the entire picture. This section discusses ontologies and ontology development. The chapter will first give a detailed overview on what ontologies are, what they are used for and how they differ from other technologies that are seemingly similar. We then discuss the methodology behind ontology development and the key aspects that surround it in order to understand the conceptualization of the FHN Ontology.

The chapter goes over the various major components of ontology development and discusses modeling and designing methodologies. We then discuss an overview of a brief explanation of CoDePs in ontology development and discuss their importance and use. Finally the section concludes with an explanation and role of Rules and Axioms in ontology development and discusses their functional importance in reasoning and performing inference on large sets of data.

2.3.1. Exploring Ontologies

Ontologies and Ontological Commitments between Systems

Ontologies are designed for the ultimate purpose of representing and defining knowledge in more structured manner that will allow for more effecting sharing and reusing of that very knowledge. An ontology is a conceptualization based specification of classes or things [50], which contains detail descriptions of certain concepts via relations and hierarchies of sub-concepts and entities [51]. With sharing and reusing of knowledge as a major factor, ontologies are a formal definition that allow for ontological commitment between different vocabularies so as to enable these vocabularies to use common definitions while participating

in an application together.

In the realm of computer science and artificial intelligence, an ontological commitment is an agreement between systems to use shared vocabulary in a coherent and consistent manner within a specific context [52]. Thus, ontological commitments play an important role in Linked Data initiative within the semantic web. Ontologies provide the definitions and set of concepts that can be referred to by systems that have these ontological commitments. Ontologies allow systems to participate together using the same knowledge base and connect to even other ontologies for even more common definitions between the participating systems. Ontologies play a critical role in sharing of knowledge between information systems. Without a definition of some vocabulary, which is provided by an ontology, it is not possible to share knowledge among different agents. Ontologies provide a common knowledge base for a system to work from. They can, in a way, be looked at as a common shared section of a brain between systems. The common section can be divided into various sub sections or domains. These domains can specialize in any field ranging from knowledge representation of medicine to representation of networking systems.

Ontologies and their Use Cases

It is now clear to us that ontologies play a critical role in the supporting information systems or software agents in sharing and reusing domain knowledge. However, there are more explicit uses of ontology development that needs to be pointed for a better understanding of their overall use. Below is a list of reasons that lead to development of ontologies [50] [51]:

- A need for reuse of domain knowledge
- A need to separate domain knowledge from operational knowledge

- Requirement to have domain assumptions explicit
- Requirement to share formal definitions and vocabularies when describing a concept
- A need to analyze domain knowledge

While ontology development has the above generalized uses for various applications, they are being used to improve web applications for specific uses. Ontologies extend web-based applications to enable new features and applications of the web. Below is a list of representative use cases of web ontologies that serve as an example [53].

Web Portals: Web portals can garner extreme uses from using ontologies. Defining an ontology can allow the community to describe their content with strong syndication by using terms and axioms that are defined within the ontology. Using the ontology will allow for describing relationships between various content that is posted on the portal as well as external web-portals. With a growing collection of data, a web portal using ontologies can use facts and definitions to make inferences that will allow for a more sophisticated system of information and knowledge retrieval.

Corporate Web Management: Larger corporations usually have multiple websites that deal with different elements of the corporation such as various product offerings & sales, press releases, research and case studies, corporate procedures, white papers etc. Currently most corporations are using simple taxonomies to organize all this variety of information. However, taxonomies lack the structure, semantic knowledge representation and inference power that multiple ontologies can offer. Multiple ontologies working together simultaneously can allow for more richness of information retrieval. This will allow for strong querying that can be performed from all aspects of the corporation whether it be a sales person looking for information that crosses sales and marketing, or whether it's a hiring

technical manager looking for an employee who has expertise in two different technologies.

Agents & Services: Ontologies can provide agents with the ability to integrate and add meaning to diverse information resources. An agent could be a system that serves as a service provider or a service planner. The task of providing services in a semantic manner, or planning activities with meaning and structure will rely on the richness of the service environment being offered and needs of the user. Domain ontologies are crucial for these service agents to be able to represent terms from various domains such as restaurants, hospitals, hotels, tourism activities etc.

2.3.2. The OWL Language.

The Web Ontology Language (OWL) language is the leading language used by the Web Semantic community to build Ontologies. During the implementation of an Ontology, it is crucial that the language be able to be processed by machines. This is perfectly done by the OWL Language and is supported by construction and editing tools such as Protégé' [19]. Protégé' is a tool that allows for ontology conceptualization, formalization and implementation phases. It also allows for evaluation by checking for inconsistencies, observing restrictions and exceptions and querying of the ontology.

OWL is basically an extension of RDF and RDFS and its primary goal is to add the expressiveness and reasoning power of description logic to the semantic web. However, it is crucial to note that not everything from RDF can be expressed similarly or at all in description logic. Some examples include expressing subclasses and certain triple expressions that cannot be expressed in description logic (DL). In order to partially overcome these issues and to allow layering within the web ontology language, three sublanguages have been defined under

OWL.

OWL-Lite: The first layer or sublanguage of the Web Ontology Language is OWL-Lite and can be used when the developer wishes to express a taxonomy with simple constraints such as cardinality of 0 and 1. OWL-Lite, as the name suggests, is the simplest OWL language and corresponds to SHIF(D) description logic [54]. OWL-Lite abides by all the restrictions OWL-DL puts on the use of OWL language, but in addition also places further restrictions by forbidding use of various other properties such as owl:oneOf, owl:UnionOf, owl:hasValue etc [56].

OWL-DL: Owl-DL is intended to support all of the description logic capabilities, thus allowing maximum expressiveness while still retaining computational completeness and decidability. OWL-DL corresponds to description logic – SHOIN(D). OWL-DL has more expressiveness than OWL-Lite and allows for all OWL language constructs. However, these constructs have certain constraints on them. For example, OWL-DL can have subclasses of classes (unlike OWL-Lite), but it cannot have a class that is an instance of another class [55].

OWL-Full: OWL-Full has no expressiveness constraints whatsoever, however it also loses computational guarantee and decidability. OWL Full is used by developers who want the maximum expressiveness of OWL that's already provided by OWL-DL, but also require the syntactic freedom of RDF with no computational guarantees. In OWL-Full, a class can be treated simultaneously as a collection of individuals and as well as an individual in its own right.

As mentioned earlier, the three sublanguages allow for layering within the OWL-language. The layering is such that every legal OWL-Lite ontology is also a legal OWL-DL ontology, and every legal OWL-DL ontology is a legal OWL-Full ontology. Hence, every legal

OWL-Lite ontology is also a legal OWL-Full ontology. The same also holds true for their OWL-conclusion counterparts. For example, every OWL-DL conclusion is a valid OWL-Full conclusion. However, the inverses of these relationships do not hold true. It is also now important to restate the fact that OWL is syntactically embedded into RDF. There every OWL ontology is essentially a valid RDF document, but the inverse does not hold true. All RDF documents are not valid OWL documents.

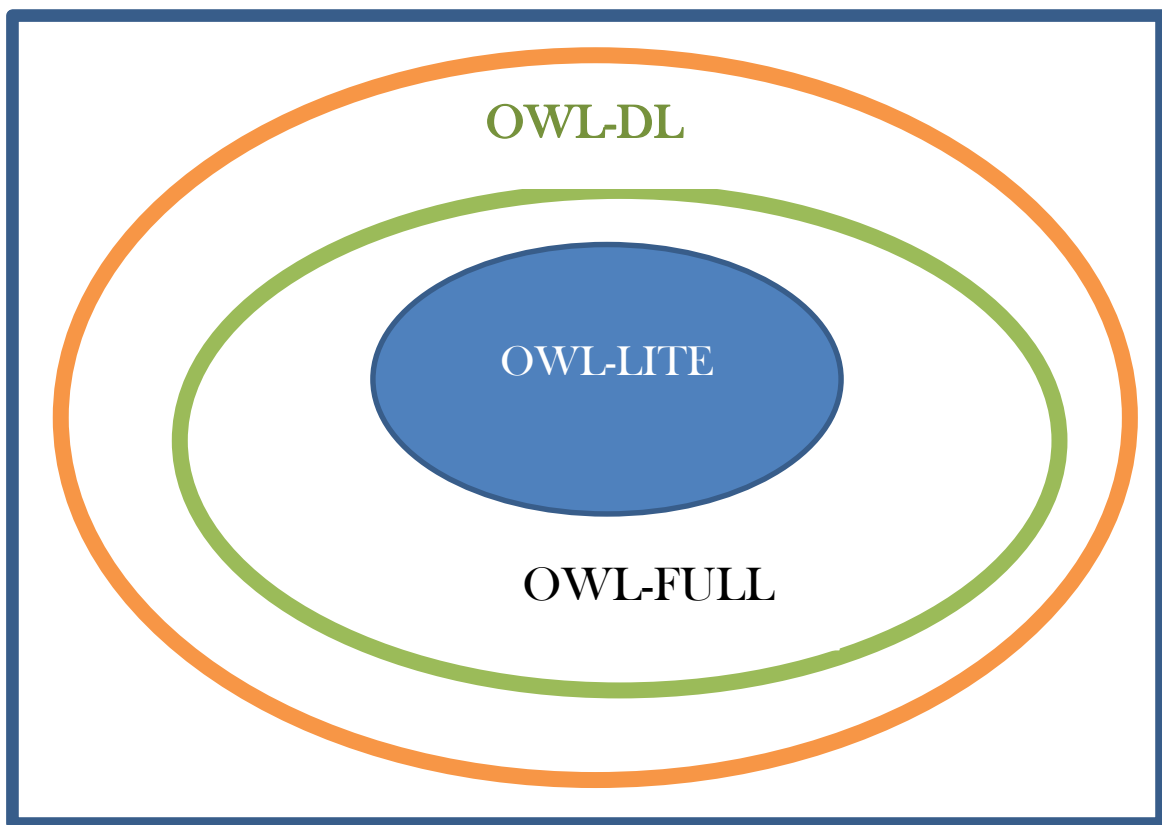


Figure 2.5: OWL Sub Languages

2.3.3. Definitions & Key Terms

There are several varying definitions of an Ontology, especially in the realm of Artificial Intelligence [15]. The term Ontology has its origins in Greek Philosophy and relates to the systematic logic of the being. In the computing world, the term is used as the sharing of knowledge in a systematic way [16]. Ontologies allow for knowledge representation for intelligent systems to infer from and share data [18]. Gruber defines an Ontology as an explicit specification of conceptualization [17].

The “conceptualization” entails developing a formal description of concepts in a domain of discourse. The components of these concepts consist of important resources like a vocabulary; which consists a list of terms within an domain that are semantically independent from reader and context, a hierarchy of concepts known as a taxonomy; and the ability for sharing and reutilization of knowledge. The major components of an Ontology are highlighted below:

Individuals: Individuals basically represent the objects within the domain in which we are interested. This is also known as the “domain of discourse”. They are instances of “classes” and “properties” may be used on them to explain the relationship between one individual or another. Classes and properties are defined below

Classes: Classes can be looked at as a “set” of individuals. A class is described using a formal description that states exactly what requirements must be met for an individual to qualify as a member of the “set” or class. A taxonomy of an Ontology comes into play when classes are organized in a superclass-subclass hierarchy. A subclass is a specialization of a superclass.

Properties: Properties allow for a binary relation between individuals. A binary relation is a relation between two things. As mentioned before properties are used to describe the semantic relationship between two individuals. Properties can have various characteristics to further emphasize the relationship between individuals. Property can have characteristics that can be inverse, functional, transitive or symmetric.

2.3.4. Ontology & Semantic Web Tools

With increasing attention toward the Semantic Web Development, there have been several XML based ontology languages that have been developed and some that are still under development. Ontology Exchange Language, XOL, is one such XML-based ontology language that was initially designed for exchange of bioinformatics ontologies. However, it can be used for ontologies in any domain [48]. SHOE (Simple HTML Ontology Extensions) is another XML-compatible extensions to HTML which allows web page authors to annotate their web documents with machine readable, or ontology level knowledge. SHOE allows for “real intelligent agent software on the web” [49].

Most importantly, the World Wide Web Consortium has made major contributions through the development of the Resource Description Framework (RDF) and The RDF Schema (RDFS). An example of continuing development of Web Semantic Tools is DAML+OIL (Darpa Agent Markup Language + Ontology Inference Layer) which is an extension on the works of the W3C's RDF and RDF-Schema with the aim to extend the languages to allow for richer modelling primitives. This section will discuss the important tools, languages and API's used in the development of ontologies and mainly in the development of the FHN Ontology. Another important contribution to the Semantic Web is the Ontology Web Language (OWL).

OWL was designed to facilitate greater machine interpretability of content on the web that that which was supported by XML and RDF-S. It does so by providing additional vocabulary and formal semantics.

While all of the mentioned tools serve different specialized needs, they all work simultaneously to achieve the same goal of allowing web content to be linked, and organized and structured in a relevant fashion. These tools combine together to provide a way for users to navigate data in a more intuitive manner. To understand how various tools and technologies come together to bring forth the Semantic Web, it is important to understand the Semantic Web Stack.

CHAPTER 3

Engineering the FHN Ontology

In this section we discuss the actual development and conceptualization of the FHN Ontology Model. Developing and engineering an ontology can be approached through various methodologies. Some engineers choose a structured approach, while others choose an unstructured approach which, to some, may seem chaotic. There isn't a "correct" way to approach the engineering of an ontology. For the most part there is no standardization of ontology development.

The absence of a well-defined design criteria, techniques and tools all contribute toward the claim by Asuncion Gomez-Perez, who stated that ontology development may not be as much an engineering activity as it is a development 'craft' in its current state [57]. In his book, *Ontological Engineering*, Gomez-Perez stated that for an art or craft to turn into a 'engineering', there must exist a definition and 'standardization of a life cycle' that goes from gathering and defining requirements to development and maintaining a finished product.

As highlighted in this chapter as well as in *METHONTOLOGY* [68], a crucial step to aid in the development and conceptualization of an ontology is developing list of Competency Questions (CQs) that serve as tool to determine whether or not the Ontology answers the questions it was set out to answer before being designed. This also serves as a verification and evaluation tool besides being a guideline while developing the ontology.

Finally we summarize the conceptualization of the classes, entities and various rules

developed to complete the FHN Model. Manfred Max-Neef's Fundamental Human Needs model encompassed theories ranging from psychology to economy and sociology. Conceptualizing the FHN Model required understanding the core concepts and definitions and most importantly, it required grasping what Max Neef set out to do in his works toward human scale development.

3.1. Knowledge Base Systems Vs Ontologies

Requirements gathering and defining is the common thorn in the side to both - Knowledge Engineers (KE) and ontologists. Requirements play a key role in specifying how the system of concern will behave. The approach towards requirements that KEs use is an incremental development approach where requirements are added to each prototype as deficiencies are found. However, this approach does not work when pertaining to ontology development. Ontologies are built with the mindset of reusability and shared linking of data which can be used anywhere and anytime, independent of the domain of the application that uses them [57]. Hence, ontologists are expected to develop an ontology that, at least partially, meets a big portion of the domain in concern. However, while ontologies require a larger portion of the domain to be covered, the domain is a lot more abstract and limited as opposed to that of a Knowledge-Based-System which can be quite comprehensive. Hence, for this reason, engineering methodologies used to develop KBS, cannot be mapped direction toward the development of an ontology.

In his book, *Ontological Engineering*, Asuncion Gomez-Perez presented a structured methodology for ontologists who wish to build an ontology from scratch. Gomez-Perez, in the chapter “Methontology – From ontological art toward Ontological Engineering”, detailed the structured methodology of Methontology that includes a set of activities and techniques to achieve the said activities [68]. The FHN Ontology was engineered and developed with the Methontology in mind. The following sections will detail the various components, techniques and aspects of the methodology used to develop the FHN Ontology.

3.2. Methontology

Asuncion Gomez-Perez presented the Methontology in order to provide ontologists with a structured overview on how to build an ontology from scratch. Using the methodology presented in Methontology [68], the first step in developing the FHN Ontology was to develop a specification document written in natural language. The specification document was constructed with help of Competency Questions that is discussed in the next section. The competency questions allow for narrowing down on the scope of the ontology and further help with the knowledge-acquisition phase of the methodology.

Defining the scope of the ontology is crucial and plays a key role in the development process. To do this we explored answers to questions like ‘What is the domain of discourse that the ontology will cover?’, ‘What is the purpose of the ontology?’, ‘What questions should the Ontology provide answers for?’ A scope of an ontology is best defined within the *specification document* of the FHN Ontology. Developing the Competency Questions (CQ) that the FHN Ontology System should be able to answer is also a significant component. In developing the Competency Questions and the specification document, an iterative approach

is adopted.

The next component of this methodology is the Knowledge Acquisition phase. Although, it must be noted that even though the Knowledge Acquisition its own individual component, it still coincides with other components. For example, knowledge acquisition is done simultaneously during the construction of the specification document. As the ontology development process moves further forward, the knowledge acquisition process decreases.

The Conceptualization Phase is the next phase in Methontology. In this component, we structure the knowledge acquired to develop the FHN ontology in a conceptual model that is able to answer the questions posed by the Competency Questions document and the specification document. The conceptualization phase includes grouping of concepts (entities) and verbs (properties), and these groupings to develop the ontology.

The final two components consist of Integration and Implementation. The Integration phase involves reuse of existing definitions and ontologies that are preexisting. This allows for speeding up the ontology development process when possible. This can be done by either analyzing existing meta-ontologies or using CoDePs (Conceptual Ontology Design Patterns). Usage of CoDePs is discussed in a preceding section. The implementation phase is self-explanatory and consists of using an environment that will allow for the development of the selected conceptualization and the chosen CoDePs or meta-ontologies. In the following sections we discuss in further detail, how each component of the Methontology was used in the process of engineering the FHN Ontology.

3.3. Competency Questions

The first phase of Methontology is best achieved by developing a list of competency

questions. Competency Questions (CQs) are used as a tool to help determine the scope of the ontology. CQs are a sketch of a list of questions that the ontology should be able to answer [26]. Once the questions are developed and sketched out, they later serve as a litmus test. The list of CQs tests whether the ontology has enough data, level of detail or particular representation to serve the needs of the intended use of the ontology. CQs can be developed based on the aim of the ontology designer. Ontologies that focus more on the domain will have more specific questions that will test whether the ontology has the correct answers to the question.

An ontology designer or an ontologist who is developing an ontology on a more abstract or higher level will focus on whether the ontology has the means to answer the questions given the right domain. In a sense, they focus more on the conceptualization of the classes and properties. In this case, the ontologists aim toward developing an ontology that will best allow for future developers and users to present a knowledge base that will conform to the ontology at hand.

Formulating and Validating the Competency Questions

In developing the FHN Ontology a similar approach was used to develop a sketch of CQs that tests the ability to answer general questions on determining Satisfiers for various needs, roles and agents. The CQs also allow for narrowing down on the scope as well as evaluation of the ontology. Coming up with the competency questions was a combination of brainstorming with a room full of people who understood the purpose of the ontology, while simultaneously consulting with Manfred Max-Neef's *Human Scale Development* book.

The conjunction of the two allowed use to narrow down on the core questions that the ontology should, at bare minimum, be able to answer at the end of its conceptualization.

After formulating the core questions, the team members were conferred with again, and examined against the resources to validate for approval. As such the FHN Ontology has the following Competency Questions

- 1) What are the satisfiers to a certain need?
- 2) What are the roles associated to a certain need?
- 3) Which agents are these roles assigned to?
- 4) Given a role, what are the satisfiers to this role?
- 5) Given an agent, can we determine the satisfiers to this agent?
- 6) Given a satisfier, what are the corresponding needs that are satisfied?
- 7) Given a need, what are the satisfiers (destroyers) that impair it?
- 8) What satisfiers impair needs of a certain agent?

With an initial sketch of these competency questions, and following an iterative approach, the conceptualization and design phase of the ontology begun. The CQs were kept in mind as a major design goal while designing the entities and relationships of the FHN Ontology. To fill in the gaps and to be able to answer questions that OWL relationships weren't powerful enough for, SWRL axioms were used and applied to the ontology.

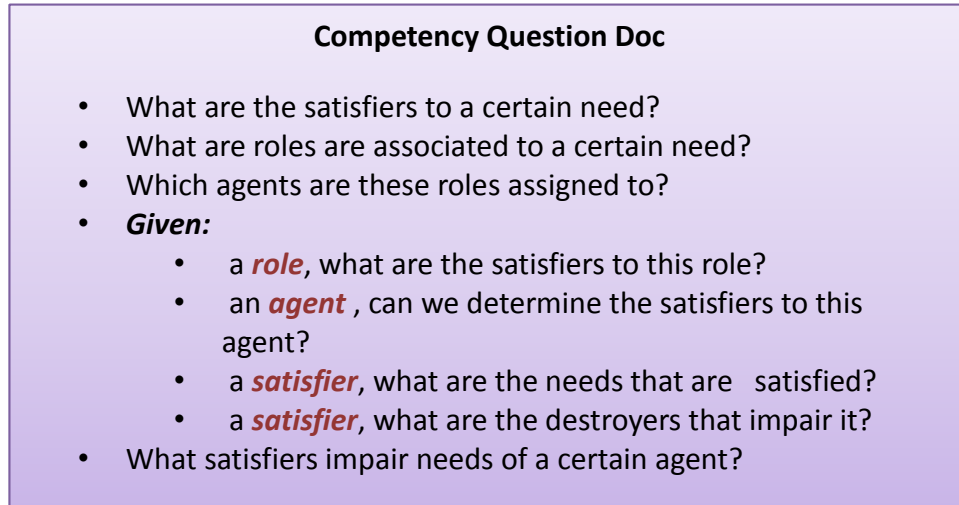


Figure 3.1: Competency Question Documentation for the FHN Ontology

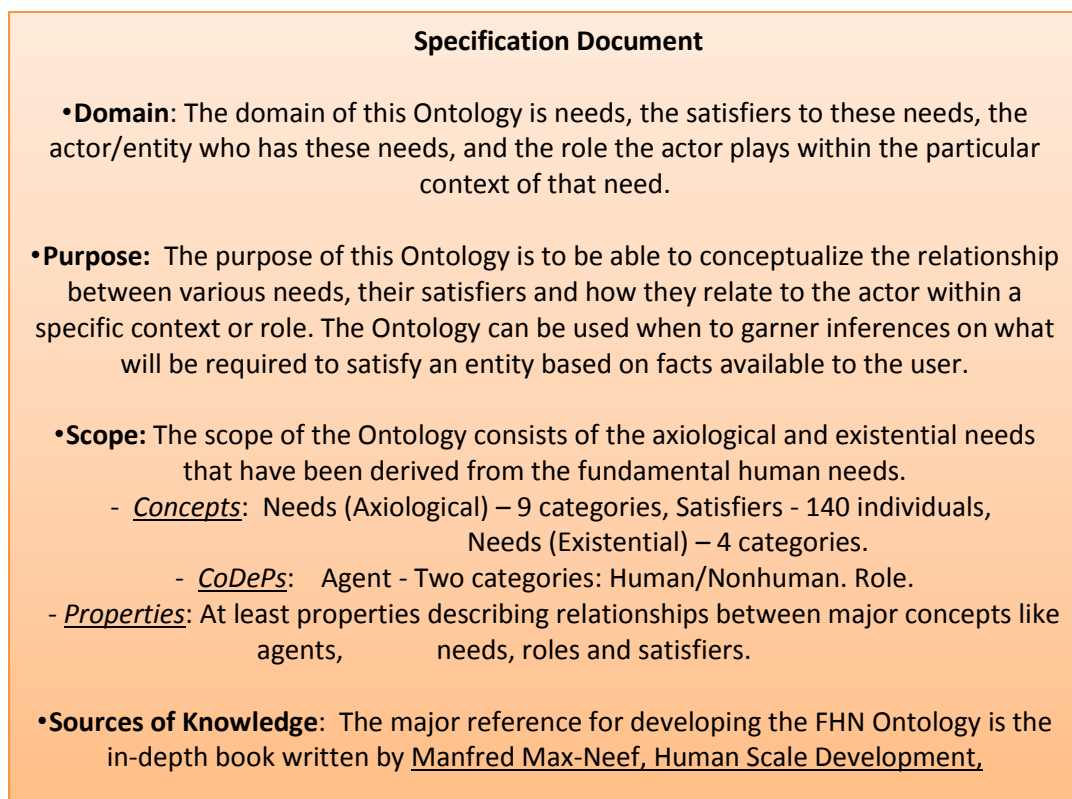


Figure 3.2: Specification Document for the FHN Ontology

3.4. Specification

As discussed in the previous section, the Ontology Specification Document is crucial in the process of engineering an ontology. A “Specification” of an Ontology is basically a requirements document that highlights the domain, purpose and scope of the ontology along with other details such as the sources of domain knowledge. A brief overview of the specification of the Fundamental Human Needs Ontology is highlighted below.

Domain: The domain of this Ontology is needs, the satisfiers to these needs, the actor/entity who has these needs, and the role the actor plays within the particular context of that need

Purpose: The purpose of this Ontology is to be able to conceptualize the relationship between various needs, their satisfiers and how they relate to the actor within a specific context or role. The Ontology can be used when to garner inferences on what will be required to satisfy an entity based on facts available to the user.

Scope: The scope of the Ontology consists of the axiological and existential needs that have been derived from the fundamental human needs.

- List of Concepts: Needs (Axiological), Satisfiers. Needs (Existential)
- Content Design Patterns: Agent/Role
- List of Properties: At least properties describing relationships between major concepts like agents, needs, roles and satisfiers.

Sources of Knowledge: The major reference for developing the FHN Ontology will be the in-depth book written by Manfred Max-Neef, Human Scale Development, where he outlines the fundamental human needs model with clear definitions, explanations and

experimentations.

3.5. Conceptualization Of The FHN Ontology

To conceptualize a needs-based ontology, we decided to use a validated human-needs model. While there are many popular models to choose from, we decided to use Manfred Max-Neef's Fundamental Human Needs. In Section 2.2 we discuss why we decided to pick the Fundamental Human Needs model. However, our work didn't end there, since the Fundamental Human Needs is merely a matrix of terms – axiological and existential that cannot be really understood by a computer. We needed to conceptualize this matrix in a more meaningful, scalable and representable manner. Furthermore, the Fundamental Human Needs only targets the describing what a need actually is and defines it as so. It does not describe its relationship to an agent or human. As such, we conceptualize the ontology to include Max-Neef's interpretation and explanation of what a human need is while also including the concepts of an agent, its role and their externalized satisfiers mapped to their needs accordingly.

During the conceptualization phase, the domain knowledge of the FHN model is structured into a conceptual model that allows the ontology to answer the questions posed in the CQ Document. The conceptualization of the FHN Ontology was an iterative process. There were several previous models that went through minor to major changes before arriving at the ontology conceptualization that we have at hand. However, a structured methodology adopted from the Methontology was adopted to arrive at the conceptualization.

The first step was to build a glossary of terms, and then group those terms into concepts that represented entities and properties that describe the relationship between

these entities. **Figure 3.3** illustrates an ontology graph that depicts the entities in the FHN Ontology with a depiction of their relationships between entities.

The solid arrowed lines represent the “has-a subclass” relationship while the dotted lines represent semantics between how the entities relate to each other. For the illustrative purposes the names of semantic relationships are not shown.

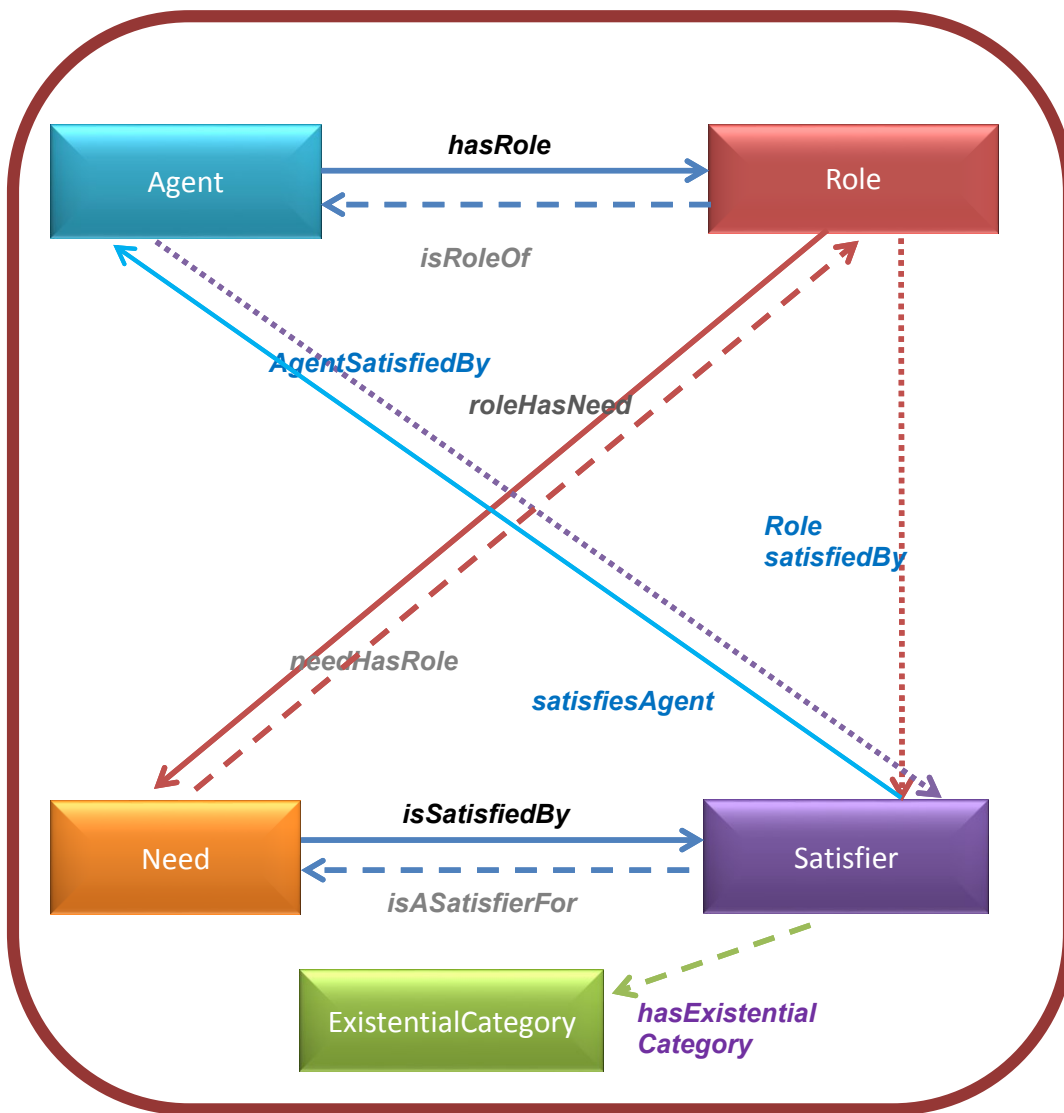


Figure 3.3: Conceptual Model of the FHN Ontology depicting major entities.

3.5.1. Entities

Satisfier: As discussed earlier in this thesis, the distinction between needs and satisfiers is of crucial importance in the Fundamental Human Needs model developed by Manfred Max-Neef. The Satisfier entity represents all individuals that would be deemed a satisfier for a certain need. This relationship is represented by the “isASatisfierFor” property between a Satisfier and a Need. The inverse is represented by “isSatisfiedBy” between a Need and a Satisfier. A Satisfier could have individuals such as: food, job, shelter, academic books, etc. **Figure 3.3** depicts the relationship between the Need entity and the Satisfier entity along with the associated relationships.

The Need entity could have individuals such as: ‘Need for Protection’, ‘Need for Understanding’, ‘Need for Subsistence’. Satisfiers and Needs can be as general as listed above or even more specific if the domain need it be. Satisfiers aren’t fixed across all uses of this ontology. Deterministic variables that the satisfiers depend on will vary from individuals, groups, and cultures.

Based on Manfred Max-Neef’s book and explanation of the relationship between a Satisfier and a Need, we can say in one statement: A Satisfier is a satisfier for at least one Need. This can be put into a predicate statement as follows:

$$\forall x \text{ Satisfier}(x) \rightarrow \exists y (\text{Need}(y) \wedge \text{isSatisfiedBy}(y,x))$$

Using the logical statement above, we can define the corresponding OWL class using RDF as follows:

```
<Class rdf:about="&FHNOntology;Satisfier">
  <rdfs:subClassOf>
    <Restriction>
      <onProperty
rdf:resource="&FHNOntology;isASatisfierFor"/>
      <someValuesFrom rdf:resource="&FHNOntology;Need"/>
    </Restriction>
  </rdfs:subClassOf>
</Class>
```

Code Snippet 1: OWL/RDF for Satisfier entity in the FHN Ontology



Figure 3.3: Conceptual model between Need and Satisfier.

Existential Category: This entity represents the categorization of satisfiers by their existential needs. Existential needs are categorized according to “Being”, “Doing”, “Having”, “Interacting”. The Existential Category has a relationship with the Satisfier entity

(Satisfier->Entity) via the `hasExistentialCategory` relationship.

Categorizing satisfiers in the according way will allow the ontology to understand whether a satisfier is something the agent should “be in a certain state to achieve satisfaction”, “have in its possession to feel satisfied”, “do in order to feel satisfied” or “be in a certain environment that will satisfy it”. For example, if a Satisfier, “academic books” has a relationship with the Existential Satisfier as “Having”, then the ontology knows that “academic books” is something the agent should have in its possession and not something that the agent should be “doing”. Existential categorization remains fixed across all uses of the FHN Ontology, but the individual satisfiers linked to this category will vary depending on the use of the ontology. **Figure 3.4** depicts the relationship between the Satisfier entity and the entity representing the existential category.

According to Max-Neef’s Fundamental Human Needs matrix, satisfiers can be categorized as either axiological or existential. We can thus say that every Satisfier has an existential category and must be either a Being, Having, Doing or Interacting subcategory.

$$\forall x \text{ Satisfier}(x) \rightarrow \exists y (\text{ExistentialCategory}(y) \wedge \text{hasExistentialCategory}(x,y))$$

Using the logical statement above, we can define the corresponding OWL class using RDF as shown in **Code Snippet 2**.

```

<Class rdf:about="&FHNontology;ExistentialCategory">
  <equivalentClass>
    <Class>
      <oneOf rdf:parseType="Collection">
        <rdf:Description
rdf:about="&FHNontology;Being"/>

        <rdf:Description
rdf:about="&FHNontology;Having"/>

        <rdf:Description
rdf:about="&FHNontology;Doing"/>

        <rdf:Description
rdf:about="&FHNontology;Interacting"/>
      </oneOf>
    </Class>
  </equivalentClass>
</Class>

```

Code Snippet 2: OWL/RDF for ExistentialCategory entity

Figure 3.4 depicts the relationship between the Satisfier entity and the entity representing the existential category. While **Figure 3.5** shows the categorization of the existential category.



Figure 3.4: Depiction of the relationship between entities Satisfier and Existential Category

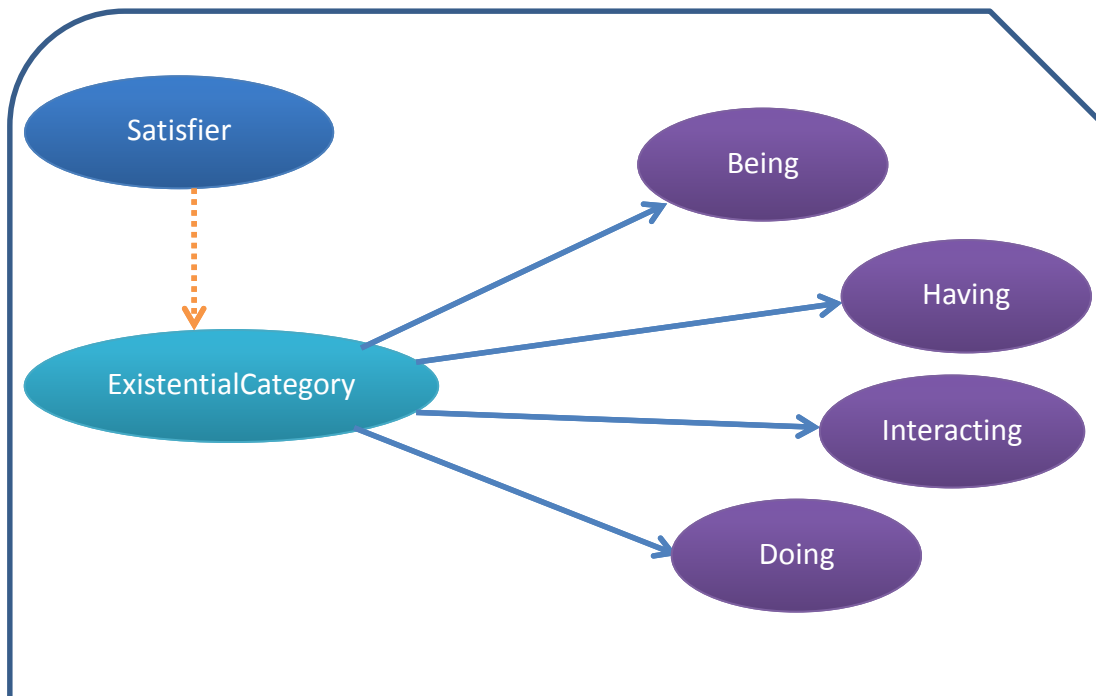


Figure 3.5: The Existential Category entity along with its associated children

Need: This entity represents the major axiological needs that were defined in Manfred Max-Neef's Fundamental Human Needs matrix. The axiological needs are defined as

Subsistence, Protection, Affection, Understanding, Participation, Idleness, Creation, Identity and Freedom. The axiological categorization, just like the existential categorization remains fixed. However, the subclasses and individuals may vary depending on the particular use for the ontology.

The Need entity is one of the most crucial entities in the FHN Ontology for the main reason that it has relationships with multiple other entities, namely the Agent entity, the Satisfier entity and the Role entity. The Need entity has a relationship with the Satisfier entity via the “isSatisfiedBy” property. The inverse relationship is represented via the “isASatisfierFor” property. The relationship with the Role entity is represented with the “roleIsNeededBy”. This triple represents the fact that the Need entity contains individuals that may be needed by certain individuals in the Role entity. The inverse of this relationship is represented by the “roleHasNeed” property. Finally, it also has a relationship with the Agent entity via the “agentHasNeed” property.

A Need is satisfied by at least one Satisfier. However, in our implementation of the FHN Ontology, we wish to state that certain roles have certain needs. This will allow the ontology to cater for multiple individuals, communities and across different cultures. Thus we say that: A need is satisfied by at least one satisfier and a need is needed by at least one role.

$$\forall x \text{ Need}(x) \rightarrow \exists y \exists z (\text{Satisfier}(y) \wedge \text{Role}(z) \wedge \text{isSatisfiedBy}(x,y) \wedge \text{isANeedFor}(x,z))$$

Using the logical statement above, we can define the corresponding OWL class using RDF as shown in **Code Snippet 3**.

```

<Class rdf:about="&FHNontology;Need">
  <rdfs:subClassOf>
    <Restriction>
      <onProperty
        rdf:resource="&FHNontology;isSatisfiedBy"/>
      <someValuesFrom
        rdf:resource="&FHNontology;Satisfier"/>
    </Restriction>
  </rdfs:subClassOf>
  <disjointWith rdf:resource="&FHNontology;Satisfier"/>
</Class>

```

Code Snippet 3: OWL/RDF for the Need entity in the FHN Ontology

Figure 3.6 depicts the Conceptual Model between the Need entity, the Role entity as well as the Satisfier entity.

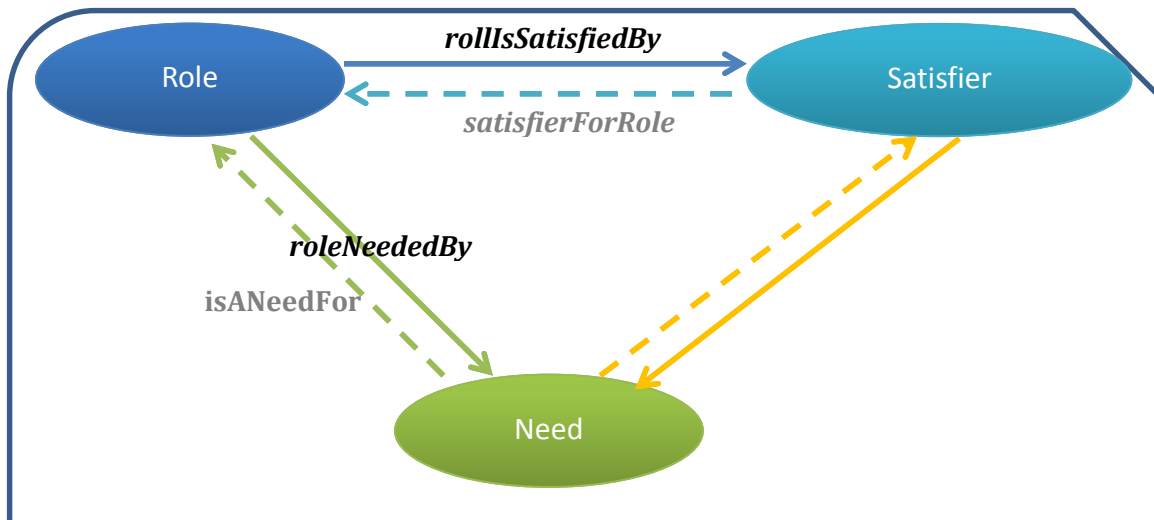


Figure 3.6: Conceptual Model showing the Need, Satisfier and Role entity with their relationships.

Role: The Role entity represents the role of a character or agent. The Role entity could have individuals such as “Professor”, “Father” , “Fitness Trainer” etc. The Role entity has a relationship with the Need entity via the “roleIsNeededBy” relationship. Consequently, the Role entity also has a relationship with the Agent entity via the “classifies” property relationship. Hence the Role entity plays a crucial entity that connects the Agent entity to the Need entity, which in turn has a relationship with the Satisfier entity.

The Role entity was added to the existing fundamental human needs model as a means for the ontology to distinguish between whom or what has a need that requires satisfaction. The Role allows the person or thing that is using the Ontology to have multiple and categorized needs. Thus, the Role has a Need and classifies an Agent. This statement can be put into logical form to allow for conceptualization as follows:

$$\forall x \text{Role}(x) \rightarrow \exists y \exists z (\text{Need}(y) \wedge \text{Agent}(z) \wedge \text{roleHasNeed}(x,y) \wedge \text{isRoleOf}(x,z))$$

Using the logical statement above, we can define the corresponding OWL class using RDF as follows:

```

<Class
rdf:about="http://www.ontologydesignpatterns.org/cp/owl/taskrole.owl#Role">

    <rdfs:subClassOf>
        <Restriction>
            <onProperty
rdf:resource="http://www.ontologydesignpatterns.org/cp/owl/classification.o
wl#classifies"/>
            <someValuesFrom
rdf:resource="http://www.ontologydesignpatterns.org/cp/owl/agentrole.owl#Ag
ent"/>
        </Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <Restriction>
            <onProperty rdf:resource="&FHNontology;roleHasNeed"/>
            <someValuesFrom rdf:resource="&FHNontology;Need"/>
        </Restriction>
    </rdfs:subClassOf>
    <disjointWith rdf:resource="&FHNontology;Need"/>
</Class>

```

Code Snippet 4: OWL/RDF of Role entity with restrictions entity

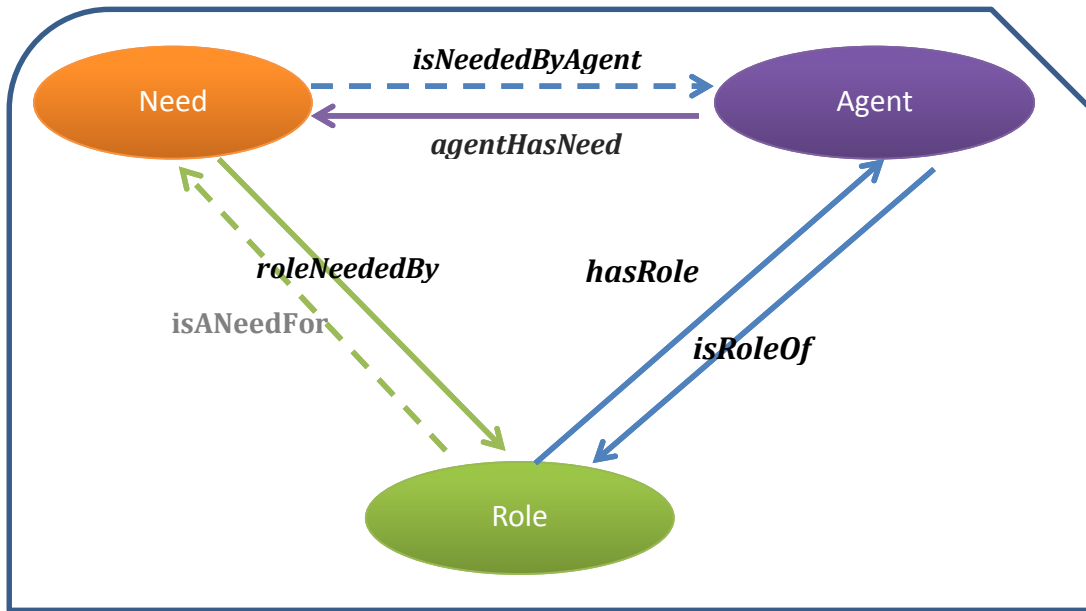


Figure 3.7: Conceptual model between Role , Need and Agent

Agent: The Agent entity is how the Ontology determines who the needs are for. An Agent is anyone or anything that may have role or a need. The Ontology will use the relationship that the Agent has to the Role entity to determine what Need the Agent has. Accordingly, with this information the FHN Ontology can determine the satisfiers that correspond to that particular need. Hence there is link formed: Agent-> Role -> Need -> Satisfier. **Figure 3.7** depicts the conceptualization of the Agent and Role entities and their relationships.

As mentioned earlier in the thesis, the Agents entity was conceptualized to be able to represent an individual or group of individuals that is using the Ontology. An Agent entity may consists individuals varying from human beings, communities to inanimate things like computers, cars etc. The Agent has at least one Role

$$\forall x \text{ Agent}(x) \rightarrow \exists y (\text{Role}(y) \wedge \text{hasRole}(x,y))$$

Destroyer: This thesis has discussed the various entities that will allow the FHN Ontology to determine the satisfiers to certain needs which correspond to certain roles and in turn correspond to an agent. The Satisfier entity thus contains individuals that satisfy the needs of an agent with a certain role. The Destroyer entity, conversely, represents individuals that impair the needs of an agent.

Each individual in the Destroyer entity impairs a need. For example, over-eating can impair the need for fat-loss. Hence, “overeating” will be an individual within the Destroyer entity. The Destroyer entity has a relationship with the Need entity much like that of the Satisfier entity, however the relationship property they share is “impairs” and the inverse corresponding relationship is “impairedBy”

According to Max-Neef a destroyer is a paradoxical satisfier that when applied to achieve a certain need, it instead impairs another need. Hence we use the statement to produce the following predicate logic:

$$\forall x \text{ Destroyer}(x) \rightarrow \exists y (\text{Need}(y) \wedge \text{impairs}(x,y))$$

Using the statement above, we can define the OWL class using RDF as follows:

```

<Class rdf:about="&FHNontology;Destroyer">
  <rdfs:subClassOf>
    <Restriction>
      <onProperty rdf:resource="&FHNontology;impairs"/>
      <someValuesFrom rdf:resource="&FHNontology;Need"/>
    </Restriction>
  </rdfs:subClassOf>
</Class>

```

Code Snippet 5: RDF/OWL for Destroyer entity with restrictions

Figure 3.8 depicts the conceptualization of the Destroyer and Need entities with the *impairs* and *impairedBy* relationship properties.



Figure 3.8: Conceptual model between the Destroyer entity and the Satisfier entity.

3.5.2. Integration of CoDePs

Methontology strongly insists on using existing ontologies in order to speed up the conceptualization and development of an ontology. CoDePs [20] [21] fit perfectly into this methodology and mindset proposed by Gomez-Perez through Methontology. Conceptual or Content Ontology Design Patterns serve as ontology-patterns that can be used within other ontologies that aim toward a broader purpose.

In Software Engineering patterns are used as a general reusable solution to problems that frequently occur within a given context in software design. A similar concept is used in ontology-design and frequently occurring conceptual problems. In ontology design CoDePs are used to fill in the conceptual representation of common tasks and relationships. Conceptual (or Content) Design Patterns (CoDePs) or Content Patterns (CPs) were hence used within the FHN Ontology to facilitate or improve techniques used during the ontology lifecycle [20].

In the simplest terms, Conceptual Design Patterns are reusable solutions to recurrent content modelling problems. They are distinguished ontologies that address a specific set of competency questions that answer the questions to the problem they provide a solution for. CPs are generally found within existing ontologies and are reused within a different domain or within the same domain and a use-case [21].

The major CODEP used for the conceptualization of FHN Ontology is the Agent – Role CoDeP used from the ontology-design-patterns (ODP) portal [58]. The CoDep fits perfectly into the conceptualization of the ontology and was designed to answer the following Competency

Questions:

- Which agent plays this role?
- What is the role that is played by this agent?

The CoDeP allows designers to make assertions on roles played by an agent without directly involving the said agent and vice versa. **Figure 3.9** below illustrates the Agent-Role CoDeP that was used in the FHN Ontology [58].

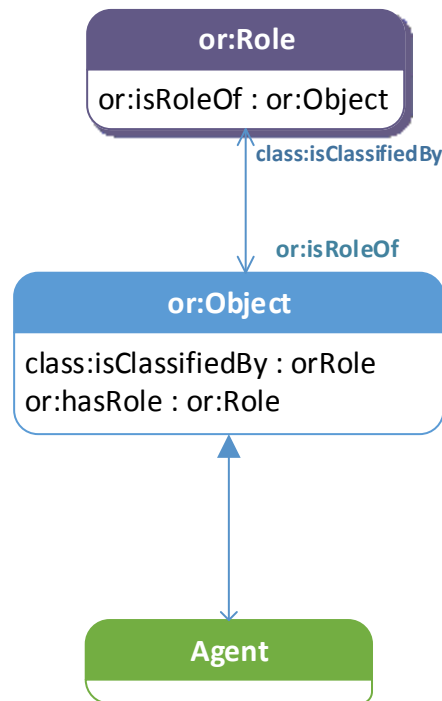


Figure 3.9: CoDeP within FHN Ontology conceptualizing Agent and Role relationship

3.5.3. Properties

Defining the properties in the FHN Ontology was especially crucial in order to conceptualize an ontology that would be able to answer the required questions. Properties allowed to define the relationships between objects and data. Furthermore, just like entities, properties have axioms called ‘property axioms’.

The property axioms used in the FHN Ontology were those from the Resource Description Framework Schema (`rdfs:subPropertyOf`, `rdfs:domain`, `rdfs:range`) and those from the Web Ontology Language (`owl:inverseOf`, `owl:Symmetric`, `owl:InverseFunctional` etc). The OWL-property axioms further extended the RDFS-property axioms to allow for better definition, conceptualization and inference power.

A brief overview of the major properties was shown in the previous section that detailed the conceptualization of the entities. In this section, we will further delve into the properties designed in the FHN Ontology by displaying the domain, range, and inverse-properties of the major properties that were specified in the Specification Document. The complete ontology consists of more properties, subproperties and property-axioms that aren’t covered in this section, but can be found in the appendices.

```
<ObjectProperty
  rdf:about="&FHNOntology;isASatisfierFor">
  <rdfs:range rdf:resource="&FHNOntology;Need"/>
  <rdfs:domain rdf:resource="&FHNOntology;Satisfier"/>
  <inverseOf
```

```

rdf:resource=="&FHNontology;isSatisfiedBy"/>
    <rdfs:subPropertyOf
        rdf:resource="&owl;topObjectProperty"/>
    </ObjectProperty>

```

Code Snippet 6: Relationship property “*isASatisfierFor*” in the FHN Ontology entities, owl-property axioms such as owl:inverseOf were used. This allowed for the inference-reasoners to answer some of the questions that were posed in the competency questions. **Code Snippet 6** illustrates a property defined in the FHN Ontology that uses the owl:inverseOf axiom. The combination of these property axioms along with Semantic Web Rules Language (SWRL) allow for the reasoner to answer every question in the CQ Document. SWRL Rules are discussed in the following section.

The code snippet above illustrates how a property is defined in the FHN Ontology using the OWL-property axioms as well as RDFS-property axioms. In the property “FHNontology:isASatisfierFor”, the relationship “isASatisfierFor” with domain and range set by the RDFS-property axioms are extended by the OWL-property axioms by defining the inverse property of this relationship via the owl:inverseOf property-axiom.

Table 2 illustrates the main properties between the major entities in the FHN Ontology. The table depicts the property relationships, along with their domain, range and InverseOf axiom properties. This table served as a look-up tool when implementing the conceptualization of the FHN Ontology.

Property Name	Domain	Range	InverseOf	Tran/Fun
<i>agentHasNeed</i>	Agent	Need	<i>neededByAgent</i>	F/F
<i>agentIsSatisfiedBy</i>	Agent	Satisfier	<i>satisfierForAgent</i>	F/F
<i>hasRole</i>	Agent	Role	<i>isRoleOf</i>	F/F
<i>isRoleOf</i>	Role	Agent	<i>hasRole</i>	F/F
<i>impairedBy</i>	Need	Destroyer	<i>impairs</i>	F/F
<i>impairs</i>	Destroyer	Need	<i>impairedBy</i>	F/F
<i>isANeedFor</i>	Need	Role	<i>roleHasNeed</i>	F/F
<i>isASatisfierFor</i>	Satisfier	Need	<i>isSatisfiedBy</i>	F/F
<i>isSatisfiedBy</i>	Need	Satisfier	<i>isASatisfierFor</i>	F/F
<i>isClassifiedBy</i>	Agent	Role	<i>isRoleOf</i>	F/F
<i>isExistentiallyNeeded</i>	Satisfier	existentialCat	<i>existentiallyCategorizes</i>	F/F
<i>roleHasNeed</i>	Role	Need	<i>IsANeedFor</i>	F/F
<i>roleIsSatisfiedBy</i>	Role	Satisfier	<i>SatisfierForRole</i>	F/F
<i>satisfierForAgent</i>	Satisfier	Agent	<i>agentIsSatisfiedBy</i>	F/F
<i>satisfierForRole</i>	Satisfier	Role	<i>roleIsSatisfiedBy</i>	F/F
<i>existentiallyCat</i>	ExistentialCat	Satisfier	<i>isExistentiallyNeededBY</i>	F/F
<i>neededByAgent</i>	Need	Agent	<i>agentHasNeed</i>	F/F
<i>inhibits</i>	InhibittingSat	Need	<i>inhibitedBy</i>	F/F
<i>inhibitedBy</i>	Need	InhibittingSat	<i>inhibits</i>	F/F

Table 2: List of properties with domain & range within FHN Ontology

3.5.4. SWRL Rules

The FHN Ontology defines the concepts within Max-Neef's Fundamental Human Needs model along with describing their relationships with each other and their relationships with the individual entities corresponding to these entities. Therefore, it's clear that the FHN Ontology is not just a plain taxonomy or hierarchy of concepts. However, for the sake of decidability, ontologies do not allow for the level of expressiveness that is needed. Rules achieve this task far better. The FHN Ontology is thus further extended by a set of rules that are based on logic programming. In this section we provide with a brief overview on the SWRL rule-based language and discuss how it was used to extend the FHN Ontology.

Rule-based systems and languages are widely used in many business applications such as process control, diagnostic fact finding and compliance monitoring. With the advent of the Semantic Web and one of its primary goals being interoperability, there has been significant recent interest in standardization of rules to allow for interoperability [25]. In the FHN System Web Stack (discussed in section **3.6.1**), the layer containing the SWRL rules sits on the upper layer, in-line with the ontology-layer. Hence, this layer should allow for further extension of the capabilities of the ontology. The SWRL rule-based language allows for the ability express rules such as chaining FHN owl-properties and bridging-rules that allow for reasoning within the domain and across domains, as well as mapping rules for interoperability between ontologies and other SWRL rules [27].

SWRL, a rule language for the Semantic Web, was designed as one of the key steps in the direction toward interoperability. SWRL is based on a combination of two of the OWL Web Ontology Language sublanguages: OWL DL and OWL Lite [15]. In the FHN Ontology, SWRL is used to write rules that are expressed in terms of OWL concepts to reason about the

individuals within the FHN Ontology. The rules will use the existing FHN Ontology base to infer even more and newer knowledge that otherwise could not be inferred. The SWRL specification imposes no restriction on reasoning when dealing with SWRL rules. Hence, a variety of reasoning based engines may be used in conjunction with SWRL rules stored in an OWL knowledge base [26].

SWRL uses the concept of antecedent-consequent pairs just like most other rule languages. The antecedent is referred to as the body of the rule and the consequent is referred to as the head of the rule. The antecedent-consequent pairs consist of a conjunction of one or more atoms. SWRL uses the base of OWL properties and classes to perform reasoning on OWL individuals. For instance, if one were to express the statement : “a person with a female child has a daughter” in SWRL, it would be done by capturing the concepts of ‘person’, ‘woman’ ‘child’ and ‘daughter’ in OWL. If one were to design this intuitively, the concept of “Female” would be a subclass of “Mother” while the child and daughter relationships can be expressed using OWL properties “hasChild” and “hasDaughter”. The corresponding SWRL rule would then be:

$$\text{Person} (?x1) \wedge \text{hasChild} (?x1, ?x2) \wedge \text{Woman} (?x2) \rightarrow \text{hasDaughter} (?x1, ?x2)$$

The above SWRL rule contains the antecedent and consequent and contains a conjunction of atoms. Execution of the rule would result in the individual mapped to variable x1 to have the relationship property hasDaughter with the individual mapped to variable x2.

Similarly, in the FHN Ontology, rules were formulated that capture various other expressive statements. For example, the following complex statement can be represented in

SWRL in an expressive manner. “An agent assigned to a role, where the role has a certain need, where the need in turn is satisfied by a set of satisfiers, would mean that the agent is satisfied by those satisfiers as well”

$$hasRole(?a, ?r) \wedge roleHasNeed (?r, ?n) \wedge isSatisfiedBy(?n, ?s) \rightarrow agentIsSatisfiedBy(?a, ?r)$$

Furthermore, other powerful but simple expressions are made in the FHN Ontology. For instance, to express the fact: “An individual who is an Agent but not a Human, must be an entity within the IoT” can be stated through SWRL as:

$$Agent(?a) \wedge (not(Human)(?a) \rightarrow IoT_Entity(?x))$$

A list of rules that are used in the FHN Ontology for information gathering purposes is depicted in **Figure 3.10**. It must be noted that some of the rules are listed there for test purposes and may never be used at all. However, they are left in there for illustration and testing purposes for future ontology designers to use when needed.

Rules:		⏏	⏏	⏏	⏏
Rules	+				
Man(?y), PersonAgent(?x), hasChild(?x, ?y) -> hasSon(?x, ?y)		?	@	X	O
hasSon(?x, ?z) -> select(?x, ?z)		?	@	X	O
PersonAgent(?x), Woman(?y), hasChild(?x, ?y) -> hasDaughter(?x, ?y)		?	@	X	O
hasSon(?x, ?z) -> count(?z), orderByDescending(?z), select(?x)		?	@	X	O
PersonAgent(?y), hasChild(?y, ?x), hasChild(?y, ?z), DifferentFrom (?x, ?z) -> hasSibling(?x, ?z)		?	@	X	O
PersonAgent(?y), hasConsort(?y, ?z), hasParent(?x, ?y) -> hasParent(?x, ?z)		?	@	X	O
Man(?y), PersonAgent(?x), hasParent(?x, ?y) -> hasFather(?x, ?y)		?	@	X	O
isASatisfierFor (?x, ?y), isExistentiallyNeededBy (?x, Having) -> Having (?x)		?	@	X	O
Role (?r), roleIsSatisfiedBy (?r, ?s) -> RoleWithSatisfiers (?r)		?	@	X	O
Need (?n), isSatisfiedBy (?n, ?s) -> NeedWithSatisfiers (?n)		?	@	X	O
isASatisfierFor (?x, ?y) -> SynergisticSingularSatisfier (?x)		?	@	X	O
Agent (?a), (not (AgentWithSatisfiers))(?a) -> AgentWithoutSatisfiers (?a)		?	@	X	O
Agent (?x), (not (HumanAgent))(?x) -> NonHuman (?x)		?	@	X	O
hasRole (?c, ?d), isANeedFor (?b, ?d), isASatisfierFor (?a, ?b) -> roleIsSatisfiedBy (?d, ?a)		?	@	X	O
isASatisfierFor (?x, ?y) -> Satisfier (?x)		?	@	X	O
isASatisfierFor (?x, ?y) -> isSatisfiedBy (?y, ?x)		?	@	X	O
Agent (?n), agentIsSatisfiedBy (?n, ?s) -> AgentWithSatisfiers (?n)		?	@	X	O
Role (?r), (not (RoleWithSatisfiers))(?r) -> RolesWithoutSatisfiers (?r)		?	@	X	O
hasRole (?c, ?d), isANeedFor (?b, ?d), isASatisfierFor (?a, ?b) -> agentIsSatisfiedBy (?c, ?a)		?	@	X	O
agentIsSatisfiedBy (?a, ?s), isSatisfiedBy (?n, ?s) -> agentHasNeed (?a, ?n)		?	@	X	O
hasRole (?c, ?d), agentIsSatisfiedBy (?c, ?b), isSatisfiedBy (?a, ?b), roleHasNeed (?d, ?a) -> agentHasNeed (?c, ?a)		?	@	X	O

Figure 3.10: Screenshot of the list of SWRL Rules currently in the FHN Ontology

3.6. Implementation

The final component of the Methontology is the implementation phase. The implementation phase involves picking the right environment to develop the ontology, use the CoDePs and construct the SWRL rules. For the purpose of engineering the FHN Ontology, the ontology was developed using RDFs and OWL and designed within the Protégé tool which included tools for OWL-visualization as well as SWRL rule building and editing. In this section we discuss RDF and OWL in brief and how they were used in the FHN Ontology.

In the computing world, semantics are usually expressed using two of the standardized languages, namely the Resource Description Framework (RDF) and the Ontology Web Language (OWL) [23]. RDF was developed by the World Wide Web Consortium (W3C) as an infrastructure that enables the encoding, exchanging, and reuse of structured metadata. One of the major features of RDF is its support for modular interoperability between metadata of different element sets. RDF does provides the means to publish machine -processable as well as human-readable vocabularies [22].

These features make RDF extremely apt for the developing a conceptualization of the Fundamental Human Needs tied with the Internet of Things. One of the cornerstones of the success of the Internet of Things is interoperability. Allowing IoT to be conceptualized with a model of Human Needs (such as FHN) allows for a interoperable understanding of Human Needs between devices across the internet. RDF Schema extends RDF by allowing for inclusion of relations between the RDF terms.

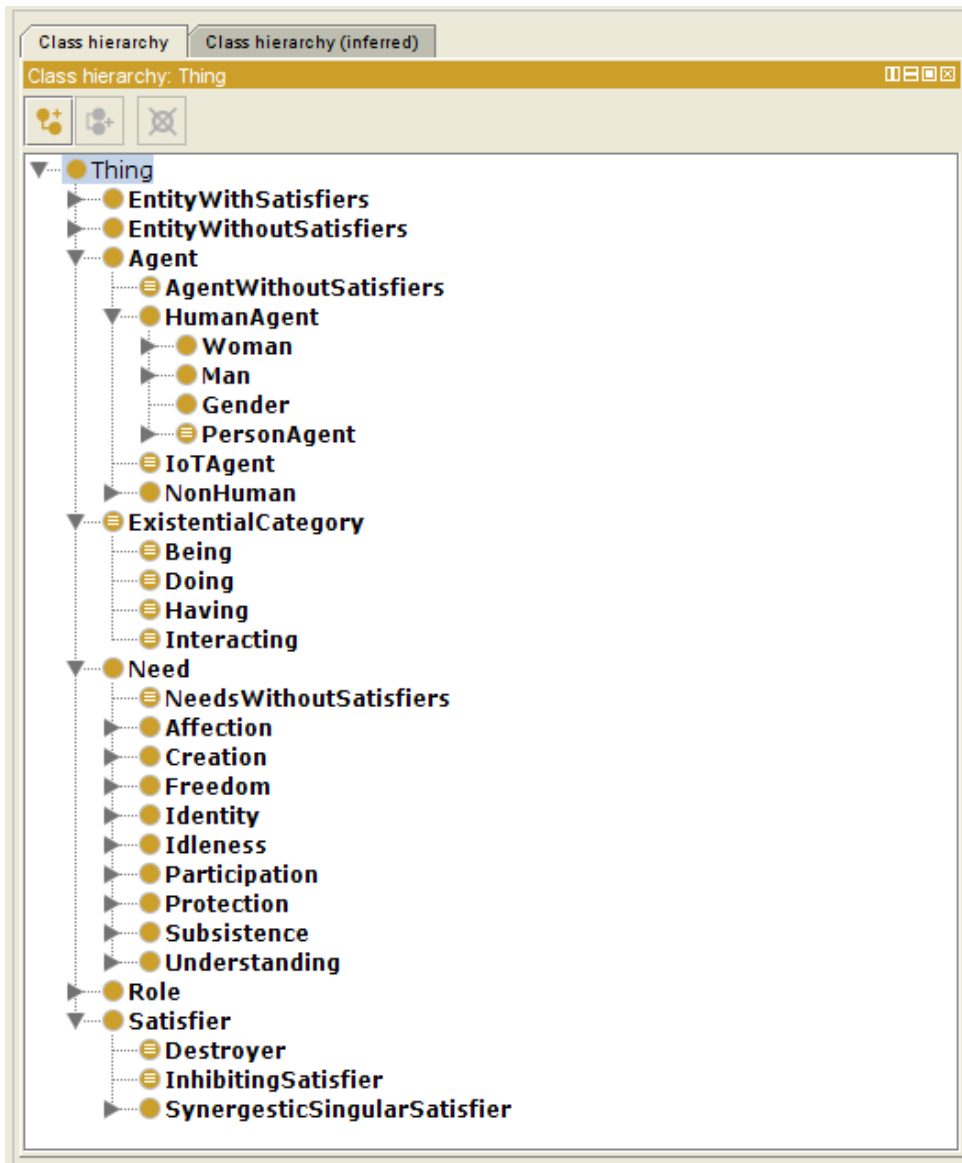


Figure 3.11: Screenshot of entity-structure of the FHN Ontology

The Ontology Web Language (OWL) is the youngest extension to the growing Semantic Web stack that's developed under the guidance of the W3C. OWL was introduced since there

was a need for machines to be able to perform reasoning task that go beyond the basic semantics that is provided by the RDF Schema [15]. The RDF Schema is powerful and allows for a description of a vocabulary that contains classes with properties and semantics for generalization-hierarchies. However, OWL further extends the stack by adding more powerful vocabulary for describing properties and classes, which in turn allow for better machine reasoning ability. Thus, the FHN Ontology takes advantage of the extended meaning and inference ability offered by OWL.

Protégé was one of the tools that aided in the conceptualization, visualization and even evaluation of the FHN Ontology. Protégé is a meta-tool and open source ontology editor used for building intelligent and knowledge based systems. Protégé allows for development of OWL in the three sublanguages of OWL, which include OWL Lite, OWL DL and OWL Full which allows for semantic expressiveness and syntactic freedom from RDF. The FHN Ontology was developed with the OWL DL sublanguage of OWL. It allows for maximum expressiveness and doesn't lose out on the computational guarantee and decidability [15]. **Figure 3.11** depicts a screenshot of the FHN Ontology entity-structure conceptualization as show in in the Protégé tool.

The Protégé Ontology development tool comes with a built in SWRL editor that allows for interactive editing of SWRL rules. The SWRL editor supports the full set of language features outlined in the SWRL Specification. The editor is tightly integrated with the Protégé development tool and developers have direct access to OWL classes, properties and individuals contained within the OWL knowledge base loaded in the tool [25].

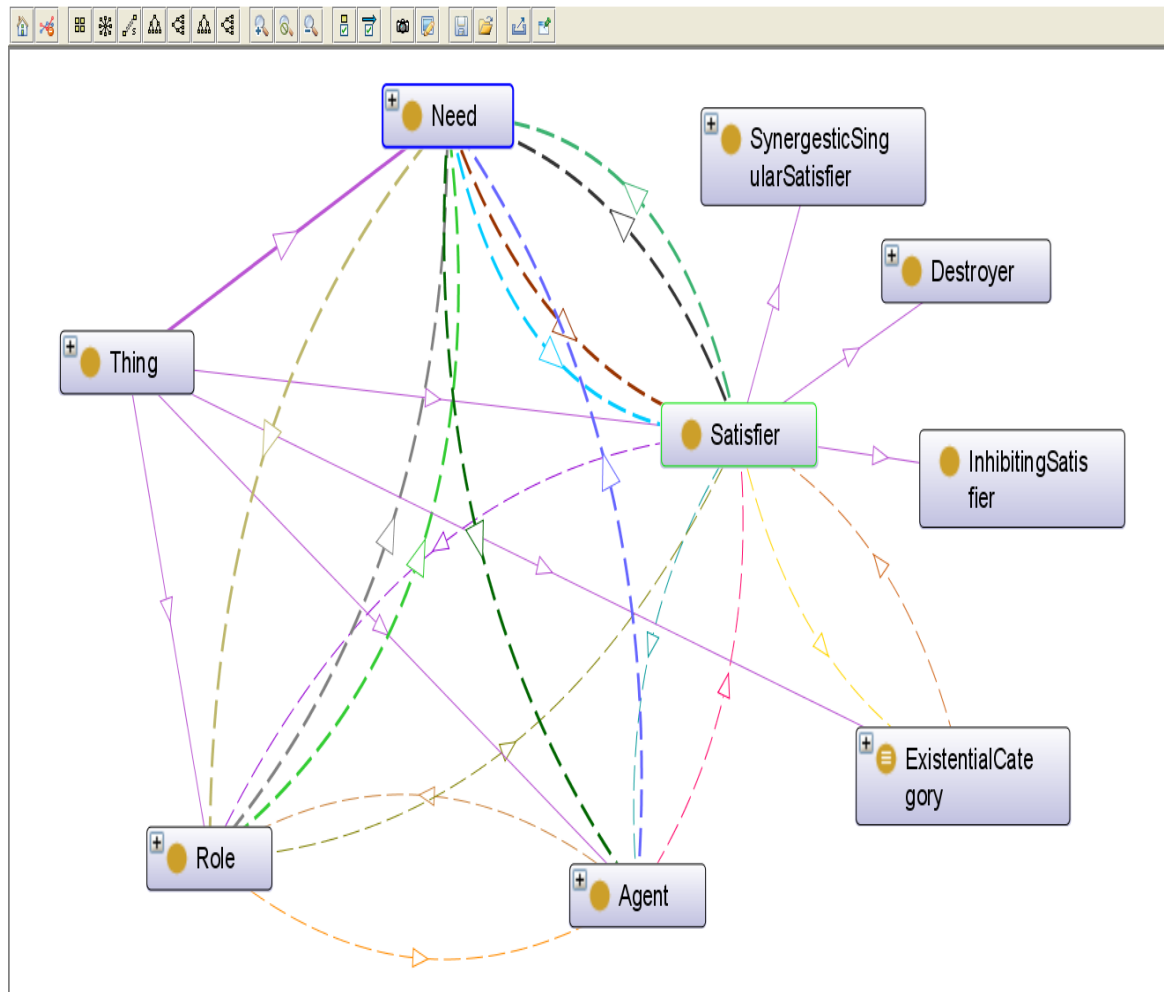


Figure 3:12: Screenshot of the FHN Ontology as visualized by the Protégé tool.

3.6.1. The FHN Technology Stack

The FHN Ontology was constructed using the Semantic Web Stack [99] as its backbone. In the previous chapters we discuss various layers of the FHN technology stack. However, we did not discuss the structure and organization of the technologies. To understand the implementation of the FHN Ontology, a look at the technology stack is warranted. In this section we will discuss a brief overview of each layer in the FHN Technology Stack. **Figure 3.13** illustrates the FHN Technology Stack.

The Semantic Web stack is continually growing layer upon layer of technologies that come together to bring forth an extended version of the web with improved data interpretability, structure and reasoning. With the Semantic Web Stack in mind, the FHN Technology Stack was chosen to appropriate its match the aims and principles while conforming to the providing a solution to the problem statement posed. It's important to note that the FHN Technology Stack is not conclusive or robust. Changes can be and are being made as time passes.

The FHN Ontology uses the general structure of the Semantic Web Stack [99] while also picking alternate technologies that fulfill similar responsibilities of that particular layer. The layer used in the FHN Ontology are listed below in correspondence to the layers in the Semantic Web stack

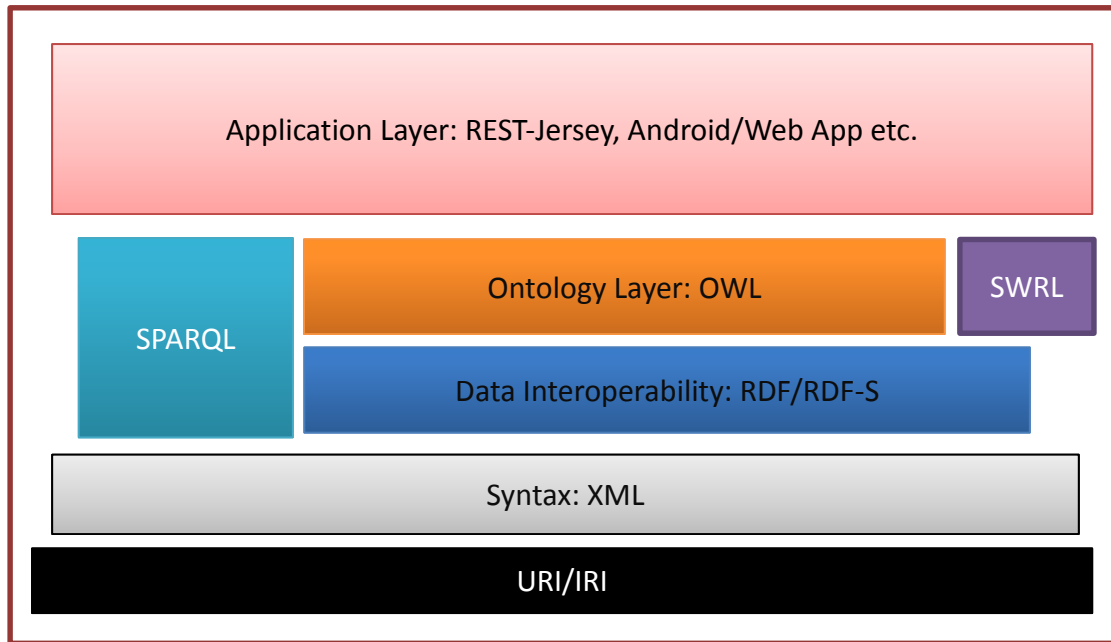


Figure 3.13: FHN Technology Stack based on the Semantic Web Stack

URI & Unicode Layer: The first layer, as with most if not all technology based on the Semantic Web Stack, consists of the Unicode and Uniform Resource Identifiers (URI). Unicode basically standardizes the encoding of international character sets, which allows all human languages to be read and written on the web in a singular standard form. The usage of URI is critical in the in the FHN Ontology, since it allows for linking to external data, thus making it more powerful and doing so in a more structured, reusable and understandable fashion. A more familiar term that is used in day-to-day life is URL – Unified Resource Locator, which is actually a subset of URI. Web documents are essentially resources, and can be accessed via the unique identifier (in this the URL). Another subset of the URI is the URN – Unique Resource Name. URNs allow for identification of a resource without implying its location.

Using the URI, the FHN Ontology is able to link to the VCard Ontology which is

externally placed. This allows the FHN Ontology to attach a profile or description to an Agent individual within the FHN Ontology.

XML Layer: The Extensible Markup Language layer consists of the XML namespace and the XML Schema definitions . This layer ensures that there will be a common backbone syntax that will be used in the FHN Ontology and other semantic web technologies. XML is a markup language that brings standardization and structure of information within a document. The namespace in XML allows the FHN Ontology to reference multiple other vocabularies in one XML documents using their URI.

RDF Layer: The RDF (The Resource Description Framework) Layer is one of the two core layers of the FHN Ontology System. This layer is responsible for separating the raw data from the actual reasoning structure from the ontology system. Hence, allowing for loose coupling, and making the system more scalable. With the RDF layer, the FHN Ontology System can have multiple data sources loaded as and when needed while not affecting or changing the reasoning layer. RDF is based on XML and hence can also reference other RDF documents, similar to how XML can do so, in a structured manner.

The Resource Description Framework, RDF, serves as a framework for representing data linking in a graph form. It's a standardized model for data exchange that also facilitates data merging even when the underlying schemas do not match. One of the key features that makes RDF crucial to the FHN Ontology System is its extension of the linking structure that the web currently holds.

RDF allows for the use of URIs to define binary relationships between objects/things, as well as identify the objects at the two ends of the relationships. These binary relationships

between objects are known as triples. The triples are what allow for seamless loading of multiple data sources into a repository since they don't need to conform to the same structural complexities of a relational database.

OWL Layer: The crux of the system lies in the OWL Layer. The Web Ontology Language (OWL) is a layer added to allow for more detailed vocabularies and ontologies. FHN Ontology system uses OWL to describe the vocabulary with greater semantics and thus allowing for stronger inferences.

The OWL language is derived from description logic and offers extended constructs over the existing RDFS features. OWL is syntactically embedded and hence is a standardized vocabulary as such. So, in essence, OWL is basically an extension of the layer that it sits over, the RDFS layer. Hence, every OWL document is in fact an RDF document, but the opposite is fallacious. OWL allows for more powerful constructs such as the “inverseOf” property that is very useful to the FHN Ontology System.

RULES Layer: The Rules layer plays a fundamental role in the FHN Ontology System in deciphering additional information by using the OWL layer and the RDF layer in combination. Since OWL and RDFS have formal semantics defined for them, it is possible for rule-engines to perform reasoning on these ontologies and knowledge bases to derive additional information. Since standardization is key in the Semantic Web, rule engines are beginning to get standardized as well and allow for defining constructs that are beyond even that of OWL. A couple of noteworthy emerging examples are the Semantic Web Rule Language and the Rule Interchange Format (RIF) that are both standardized. The FHN Ontology uses SWRL as its formal rule defining format, but also uses the SPARQL Construct to define rules for prototyping purposes within the Sesame Framework.

Application Layer: Finally, the last layer that sits on top of this stack is the Application Layer. The FHN Ontology System uses a REST-Web Service as the heart of its application layer. The Rest-Web Service acts as end-point that allows external applications to consume the API. This allows for web-applications, native mobile applications and others to connect to the end-point of the FHN Ontology and “ask” the FHN Ontology the questions they need answered. Furthermore, this also serves as a singular hub for all client-side users to easily upload their data anonymously and thus helping the strengthen the inference power of the FHN Ontology as a whole.

The REST-Web Service was implemented using the Java API for RESTful Services, also known as JAX-RS API. The framework used, that implements the JAX-RS API specification, is JERSEY. However, the Jersey Framework extends the JAX-RS API even further by providing numerous other extensions [59]. To test and evaluate the functionality of the of the FHN Ontology as well as the RESTful Service, the client-side application layer was developed in Native Android which acted as a Rest-Client and consumed the FHN-WebService accordingly. We discuss the application layer in more detail in the Evaluation Section.

CHAPTER 4

FHN Ontology System

In the previous chapter we discussed the engineering of the actual FHN Ontology. The chapter highlighted the various components of the methodology used during the engineering. These included the formulation of the specification document, the competency question document, the conceptualization of the FHN ontology mode and the implementation of the ontology. The chapter concluded with a brief overview of the FHN Technology-Stack which culminated with the application layer sitting right on the top of the technology stack. While, the underlying stack forms the crux of the FHN Ontology System, the application layer plays a crucial role in being able to use the FHN Ontology in conjunction with external data in order to prove useful in real-world applications.

For this purpose, three components were developed to showcase and evaluate the power of the FHN Ontology: FHN Data Model, FHN Rest Web Service, Native Android Client Prototype.

In this chapter, we will discuss each of the three components by providing an overview of the underlying data model and an interaction diagram that shows how the components interact with each other as well as external APIs used. We first begin with an overview of how each of the components works together.

4.1. Cloud Based System: Software As A Service

In order to contribute directly to the Semantic Web and the Open Linked Data initiative, we set out to make the FHN Ontology System available on the internet (on the cloud) and accessible via multiple applications. This makes the FHN Ontology available to devices across various application layers, whether it be a mobile application or a web browser applications. Hence, the FHN Ontology uses a combination of a domain model, a REST WebService and an external repository within which the FHN Ontology resides, in order to provide the inference power of the FHN Ontology as a service to applications developed on different environments. Furthermore, the RESTful web service also takes advantages of other cloud services for information retrieval such as Google-play-services and other ontology knowledge bases that reside on the cloud such as DBpedia to make the entire system a cloud based system.

4.2. System Components

The system comprises of three major components and supplemental components. Furthermore, an illustration of the system components can be visualized in **Figure 4.1**.

FHN Data Model: The FHN Data Model is the object-oriented model that conceptualizes the entities in the FHN Ontology programmatically in Java. This data model consists of various objects that represent individuals in the FHN Ontology. The component uses an open source external library called Sesame. This allows for parsing, storing an inferencing and querying of RDF data and serves as the major framework used to access the FHN Ontology [60].

External Sesame Library: Sesame is a Java-based open source framework that provides with the ability for processing of RDF along with other features that allow for flexibility,

scalability and optimization [60]. One of the important features of Sesame is that it can be used to connect to other SPARQL endpoints. Using this, the FHN Ontology System is able to leverage the power of the Semantic Web and linked data. Sesame provides with two ways of data access of the ontology in concern: an in-memory store and a native-store.

This grants flexibility in how the Data Model is used. Furthermore, the Sesame API supports every mainstream RDF file format, ranging from Turtle, N-Triples to TriG and TriX. This feature, in turn, gives FHN Repository to store data formats of any variety, further leveraging the power of linked-data. As such, the FHN Repo holds RDF data as well as data in Turtle format. Another noteworthy convenience of the Sesame Library is that it is available as a WAR that is tailored for quick deployment on application servers such as Apache Tomcat [62].

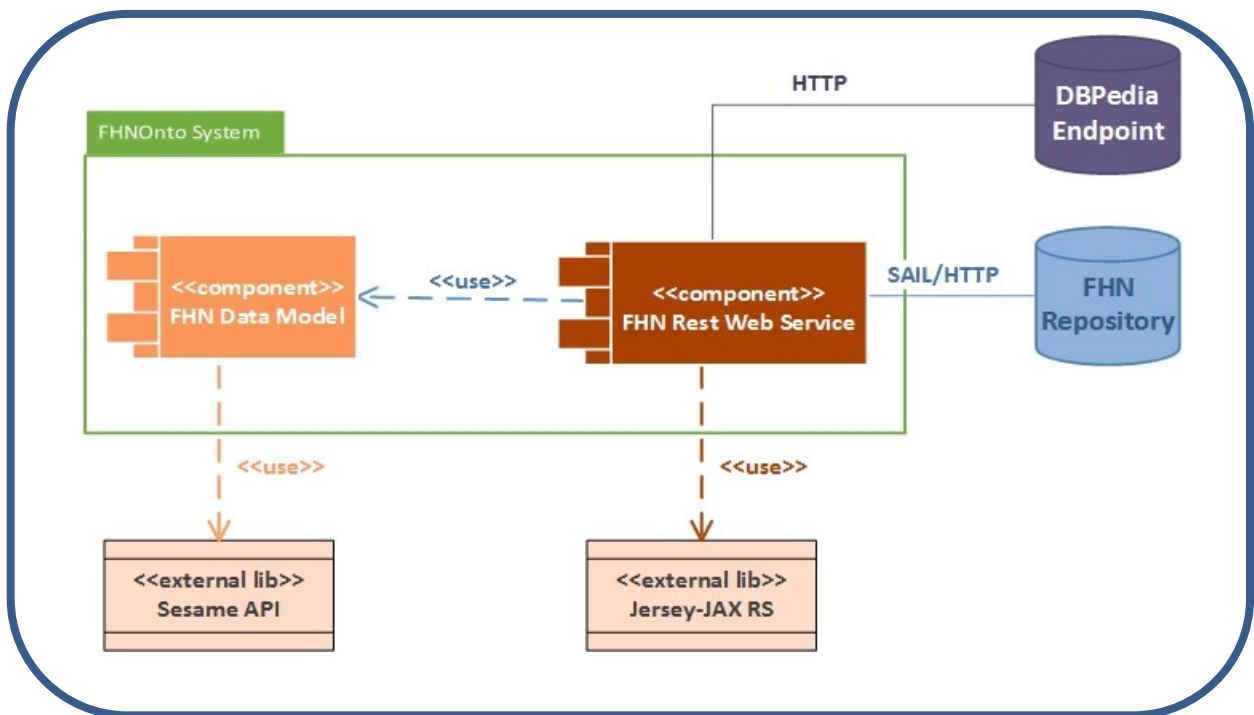


Figure 4.1: Component Diagram of the FHN Ontology System

FHN REST Web Service: The aim of this component was to be able to allow exposure and accessibility of the questions answered by the FHN Ontology in a variety of representational media types. As the name of the component implies, the FHN Rest Web Service serves as an endpoint for a client to connect to. The FHN Rest Web Service uses the FHN Data Model to model the incoming data into individual objects that map to the FHN Ontology which is stored in the FHN Sesame Repository. The FHN Rest Web Service uses the Data Model in conjunction with a query-dispatcher to query the FHN Ontology and return the response back to the client. Additionally, it also connects to the DBPedia SPARQL endpoint to make further queries and provide stronger responses to the client if and when necessary. To implement the RESTful service, the component uses an implementation of the JAX-RS API called the Jersey API [61].

External Jersey Library: Much like Sesame, the Jersey library is another java-based open-source framework. Jersey, however, is used for developing RESTful Web Services and supports the implementation of the JAX-RS API specification while also extending the JAX-RS toolkit with additional features and utilities [61] that simplifies the FHN Web Service development. This allowed for rapid building and prototyping of the FHN Restful Web Service and Client prototypes.

FHN Sesame Repository: This component represents the storage of the actual FHN Ontology. Sesame allows for three methods of ontology storage and processing: in-memory, native and accessing the Sesame OpenRDF Workbench where a native repository is stored. The method chosen for the FHN Ontology is to connect to the Sesame OpenRDF Workbench. The OpenRDF Workbench is a web-application that allows the admin of the FHN WebService to interact with the FHN Ontology within a named repository. This allows the admin to query, update and explore the FHN Ontology in a more meaningful way. Furthermore, additional

datasets in various other formats can be added to the repository which can be integrated with the FHN Ontology. **Figure 4.1** illustrates a component diagram that shows how the FHN Repository connects with the other components within of the FHN Ontology system.

4.3. The FHN Domain Model

In the previous section we discussed the major components the FHN Ontology System and how they interact with each other of. In this section we will take a closer look at the FHN Domain Model. The aim of the FHN Ontology System is to be able to capture data, conceptualize it within the FHN Ontology and use the power of OWL to provide more meaningful answers to the agent. To achieve this, a domain model was engineered to capture the data required for the FHN Ontology.

The FHN Domain Model consist of POJOs, which is a simple implementation of Java Objects that represent the necessary entities in the FHN Ontology. An illustration of the domain model can be visualized via a UML diagram which is depicted in **Figure 4.2**.

As can be seen in **Figure 4.2** the domain model represents the structure of the FHN Ontology in several ways. However, the main purpose of encapsulating the data of each individual is achieved very effectively with this model. For the purpose of brevity, we will dive deep only into the major aspects of the FHN Domain Model and discuss in brief the other aspects.

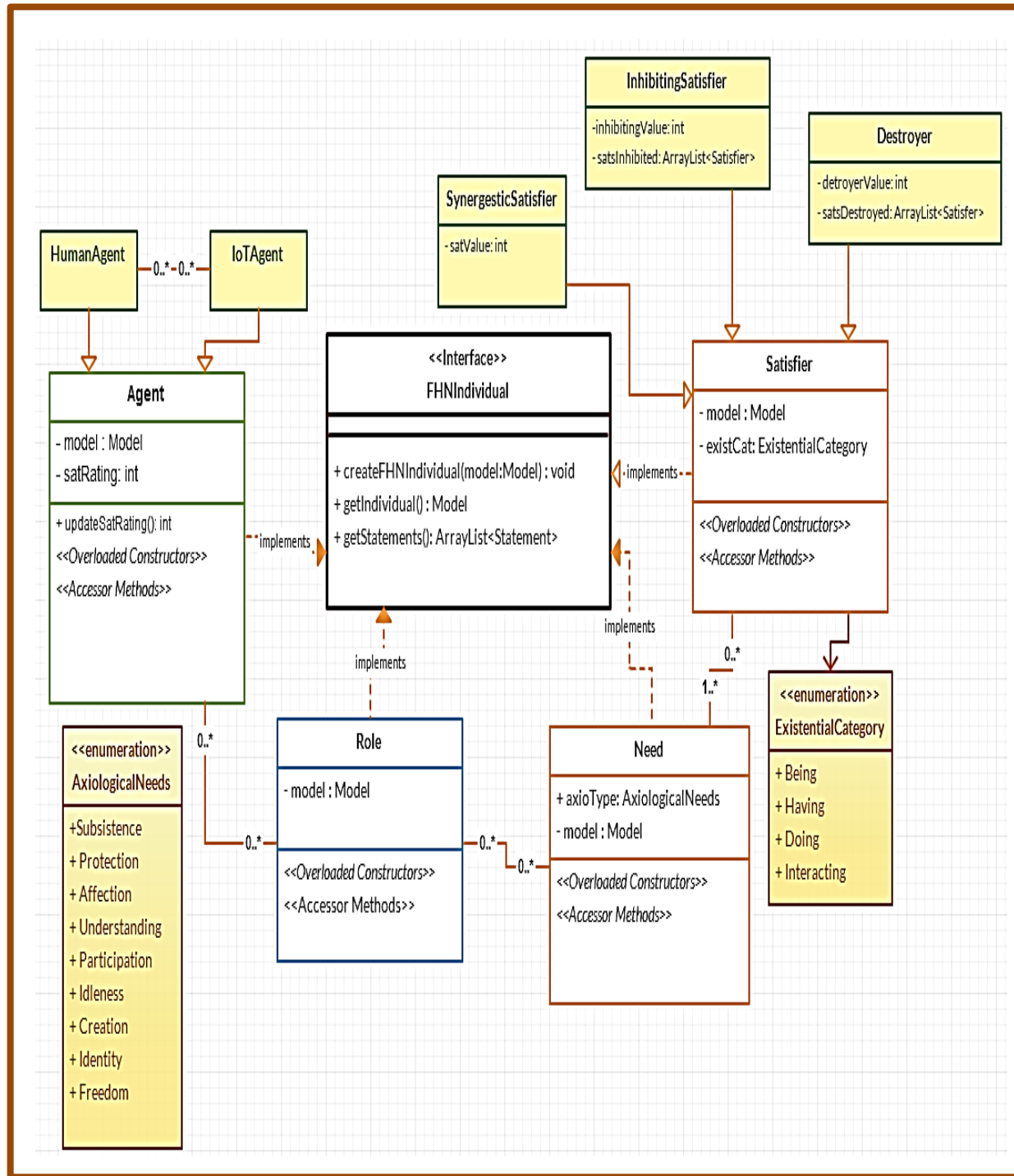


Figure 4.2: UML Diagram of the FHN Domain Model

FHNIndividual Interface: The `FHNIndividual` Interface is implemented by every object that represents the data encapsulation of an individual within the FHN Ontology. For this purpose, the `FHNIndividual` interface defines what each of the FHN-mapped POJO objects should implement. In our data model, the major parent classes that implement the interface are the `Agent`, `Role`, `Need` and `Satisfier` classes with their corresponding children classes. These parent classes and their children must conform to the minimum requirements set by the `FHNIndividual` Interface. The following methods are defined in the `FHNIndividual` Interface to be implemented.

- **`createFHNIndividual(model:Model)`** uses the `ValueFactory` and `Statement` object from the Sesame API to be added into the `Model` object. This `Model` is then used when adding, updating or deleting an individual from the Sesame Repository holding the FHN Ontology.
- **`getIndividual(): Model`** returns the `Model` that contains the encapsulated `FHNIndividual` that was created by `createFHNIndividual(model:Model)`. Each of these represent the corresponding POJO object that implements the `FHNIndividual` interface. The returned model will have access to all the namespace statements held by that individual, which can be used to add to an ontology when needed.

- **getStatements(): ArrayList<Statement>** this method will return the RDF statements that correspond to the individual object in concern. Each of the objects in the returned ArrayList is of type **Statement** which will have access to the subject (**Resource**), predicate (URI) and object (**Resource**) of the **FHNIndividual**.

Satisfier: The **Satisfier** object implements the **FHNIndividual** interface and has three children classes that extend it: **SynergisticSatisfier**, **InhibitingSatisfier** and **Destroyer**. Each of these children classes have an associated **int** value that represents a value that will be used when calling method **updateSatRating(): int** on an **FHNIndividual** of type **Agent**. Each **Satisfier** individual will have a different value assigned to depending on various factors that are beyond the scope of this thesis. Furthermore the **InhibitingSatisfier** and **Destroyer** each have an **ArrayList** that hold a list of type **Satisfier** that represent the satisfiers that they are associated to. Lastly, the **Satisfier** parent class has an **Enumeration** type that represents each of the existential categories of in Manfred Max-Neef's Fundamental Human Needs model: **Being**, **Having**, **Doing** and **Interacting**

Agent: The **Agent** object implements the **FHNIndividual** interface and has two children classes that extend it: **HumanAgent** and an **IoTAgent**. This implementation is temporary in order to conform to the existing FHN Ontology model and will be extended as the FHN Ontology is extended. Furthermore, it also has an **satRating** of type **int** that represents the total satisfaction rating of the particular agent and is called by the method **updateSatRating(): int**

Role: This is a pretty straight forward class that represents the role Entity. It has a **Model** object that holds the corresponding subject, predicate and object of the

FHNIndividual. As can be seen in the UML diagram, Role has a many-to-many association with Agent and Need.

Need: The Need object implements the FHNIndividual interface and the most noteworthy aspect here is that the axiological component of Manfred Max-Neef's categorization of needs is represented via an Enumeration type. The AxiologicalNeeds enumeration has the following elements: Subsistence, Protection, Understanding, Participation, Idleness, Creation, Identity and Freedom. Each of these categorizes a Need individual in this domain model.

4.4. Application Sequence Interaction

In the previous sections we discussed how the components structure and the domain model of the FHN Ontology System. However, we didn't cover a use-case that included the client interaction with the system. In this section we will discuss the sequential interaction beginning from the client and moving all the way to the FHN Sesame Repository and back.

4.3.1. Sequence Interaction

Figure 6.3 illustrates the sequence diagram of the FHN Ontology System. As mentioned in the previous section, the first step in the sequence is the user logging into the application. This is done by connecting the Google API via the `GoogleApiClient.connect()` which returns a `PeopleApi` object that has access to object that holds the user data from Google+. The user data is then used by the FHNReco app to create a user-object that is passed through by the RESTful template after consuming the FHN RESTful Web Service (FHNRest WS) in order to make a request to the API. The FHNAApp then awaits a response from the FHRest WS. The

web-service obtains the user-object that was passed in by the RestTemplate to create new FHNIndividual objects.

A query of type String is then created by a query-dispatcher which is subsequently used to update the FHN Sesame Repository with the new individuals. Once the repository is updated with the required individuals a series of SPARQL queries are performed on the repository to obtain the answers to the desired questions or queries by the FHNReco app. Furthermore, a second step is performed by using the answers to query a third party endpoint which allows for querying a knowledge base.

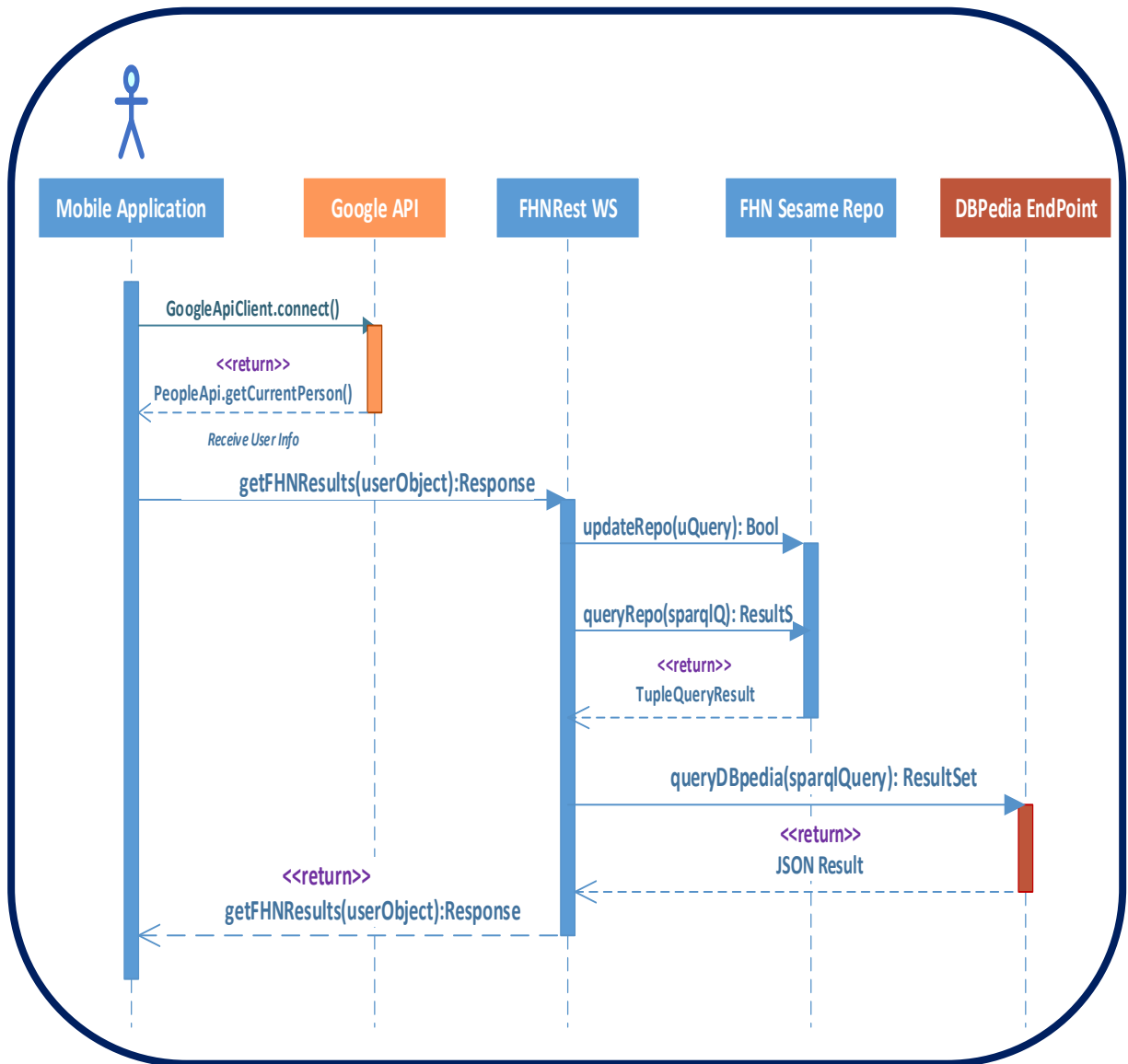


Figure 4.3: Sequence Diagram depicting flow of events from Client to Server

4.5. Testing the System

To the test the system, multiple approaches were used. Initially the system was tested to check if results could be displayed on the console. We then graduated to testing if the web-service would provide results to a simple web-based application on a desktop

environment, and finally tested if a native android application could connect to the ontology service. In this section we discuss FHNReco, a native android app that we used to depict if a user can seamlessly get a result-set of *satisfiers* from the FHNRest WS.

4.4.1. FHNReco: Native Android App

For the purpose of demonstrating a use-case of the FHN Ontology System, a client component was needed. With this purpose in mind, a native Android application was developed called the FHNReco app . The FHNReco app served as our RESTful client that consumed the FHNRest Web Service. The FHNReco app required the user to login into the application in order to identify the user and store user data.

To simplify this process for the user, the user is allowed to sign-in via their Google+ account. This allows for a quick login and also provides the FHNReco app with relevant information such as profession, age, gender , relationship status etc. Unlike Jersey and Sesame, where the FHN System uses maven to access the external JARS, the FHNReco app is gradle based and hence accesses the gradle repository to access the google-play-services api.

```
repositories {  
    mavenCentral()  
}  
  
dependencies {  
    compile 'com.google.apis:google-api-services-  
        androidpublisher:v1-rev56-1.20.0' }
```

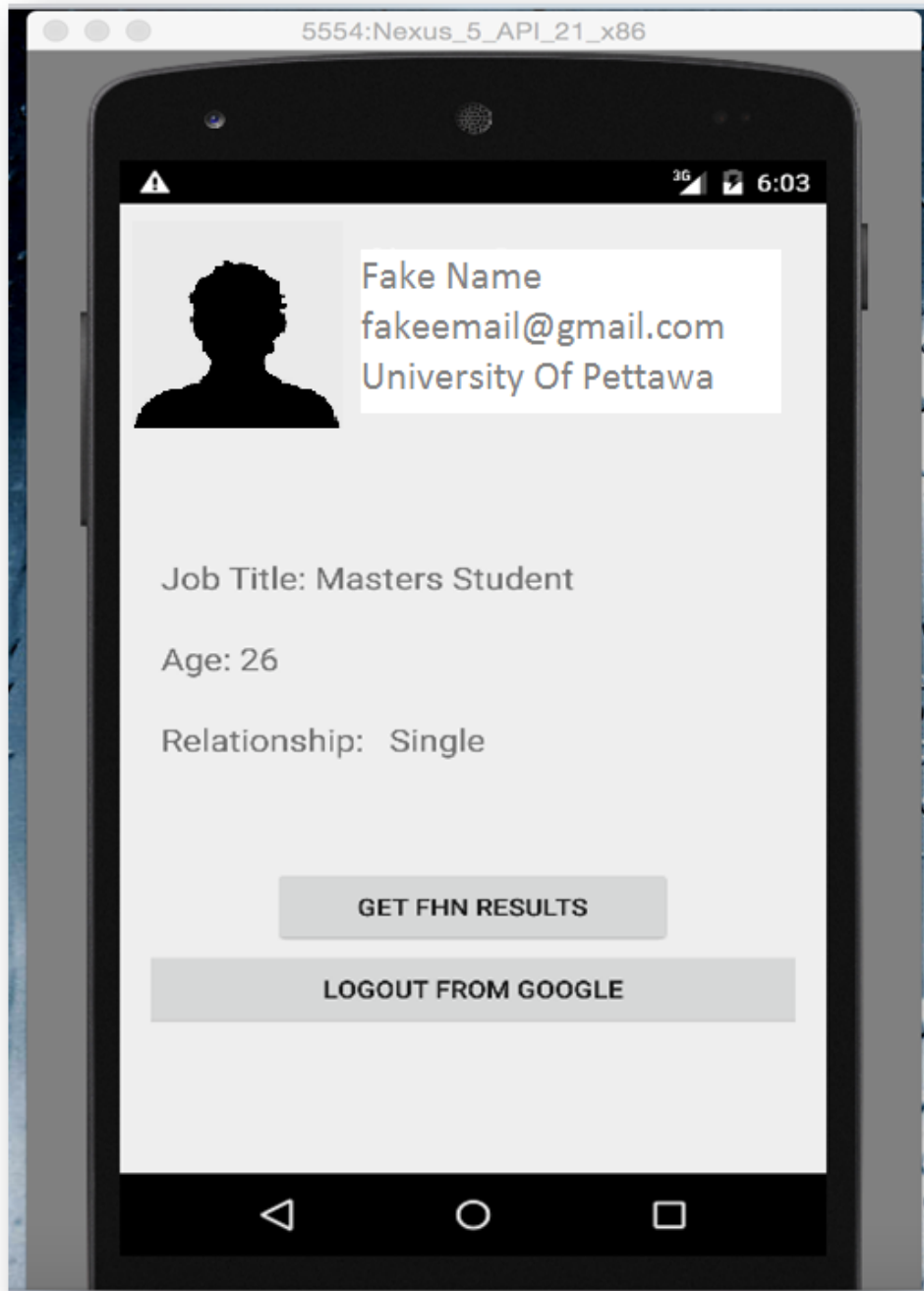
Code Snippet 7: Maven/Gradle dependency for Android App

After going through the necessary additional steps in the Google API Console [63], the application can connect to the Google+ API. The FHNAApp then creates an AgentUser object that holds the necessary user data extracted from the Google+ API. It then uses a RestTemplate object that allows it to connect to the FHN RESTful Web Service and requests for a JSON result set back. The rest of the work is done by the components discussed in the previous sections. In the next section we will discuss the sequence of events that takes place with an illustration diagram.

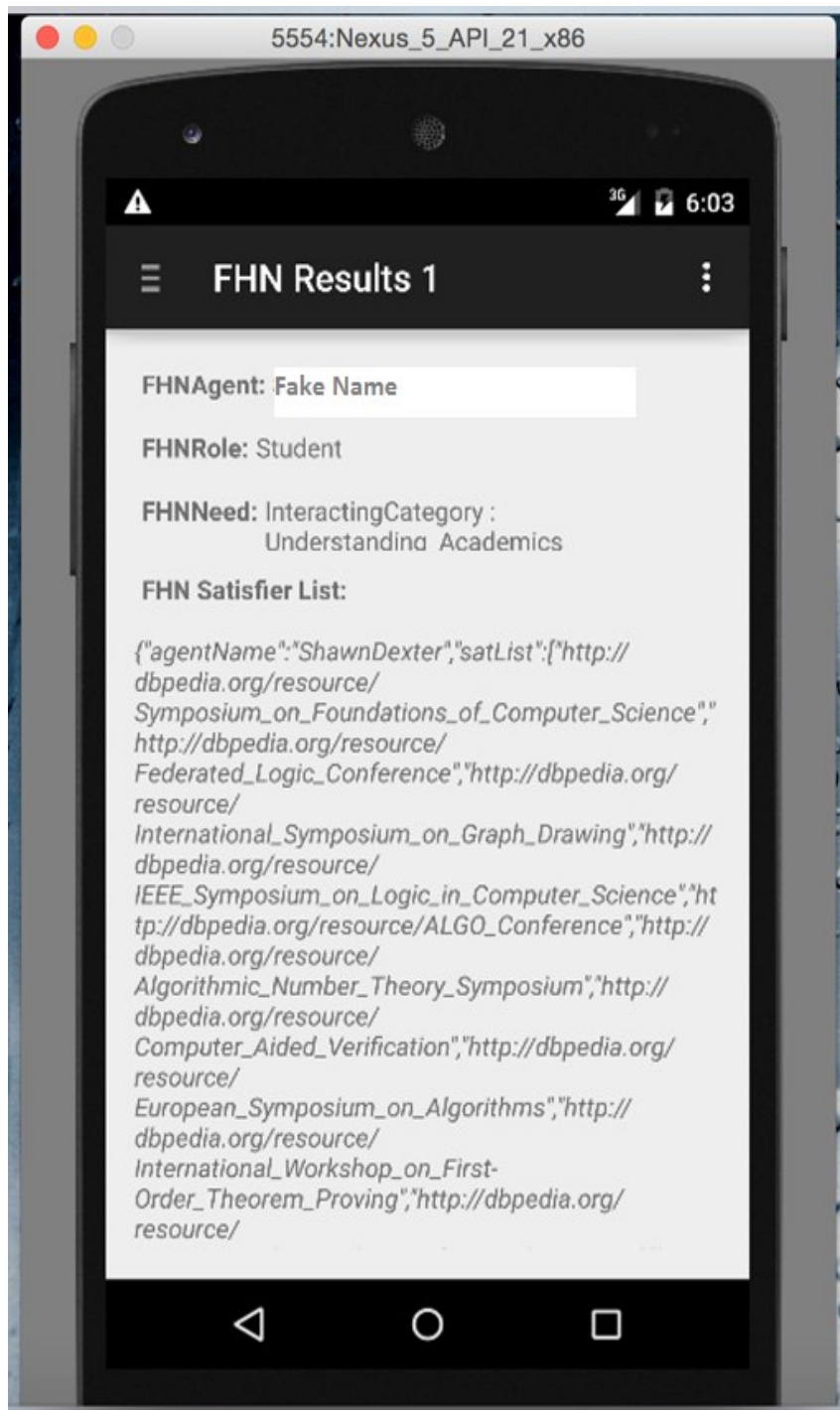
The results contain a recommendation of possible “Satisfiers” to the logged in user. At this stage of the project, the application uses current job title and the relationship status as the roles for the user. However, other profile information is used to create the Agent object in the domain model. The current state proves sufficient to serve as a proof-of-concept.

The system connects to the DBPedia endpoint. DBPedia is a linked-data and semantic web furthering initiative that extracts structured information from the Wikipedia info-boxes and makes it available for querying and for linking via linked-data [64]. DBPedia returns a result set in JSON format. These results are accordingly forwarded back to the FHNReco app as response message. The FHNReco application, in turn, displays the recommendations accordingly. The illustration in the figure above displays the sequence of events in a manner that is visually more comprehensible.

Screenshot I & Screenshot II are illustrate the prototyped native android application displaying results from the FHNRest Service. Please note that the application is currently showing the satisfiers with their full URI as it’s not formatted, the FHNNeed individual was used as a key word to query DBPedia.



Screenshot I: FHN Android App using Google+ Profile Info for creating new HumanAgent



Screenshot II: Native Android App displaying query results from FHNRest & DBPedia

The last decade has seen a strong interest toward the research on ontology evaluation methodologies [69]. The techniques of the methods on ontology evaluation range from pattern-based evaluation, to methods that check the quality of an ontology based on metrics, features and criteria [92]. Evaluation of the ontology, in fact, was the final component of the Methontology methodology proposed by Gomez-Perez. Gomez-Perez stated that in the world of ontologies, evaluation refers to the formulation of technical judgement of an ontology system with respect to a frame of reference. This technical judgment must be carried out during each phase and as well as between each phase of the ontology life cycle.

5.1. Background on Ontology Evaluation

Gomez-Perez and his colleagues note that “Evaluation” subsumes the two concepts: Verification and Validation [69]. The former referring to the technical process that aims to decipher the *correctness* of an ontology while the latter aims to guarantee that the ontology sets out to do what it aimed to do [70]. Based on the experience of verifying ontologies from Ontolingua, some ideas and guidelines on how to look for incompleteness, inconsistencies and redundancies have been presented in another paper by Gomez-Perez and colleagues [71].

In 2014, Y. Ren et al. [73] stressed the importance of evaluating ontologies via its

requirements specification by leveraging the concept of competency question and a “test-before software development process”. The aim was to automate the *validation* of an ontology by automating the process of checking that the competency questions are answered and hence the requirements are being met. A similar importance was placed on the role competency questions play in the evaluation of ontologies in 2013. C Bezerra et al. [74], in their paper “*Evaluating ontologies with competency questions*”, proposes a mechanism to support evaluating whether an ontology follows their corresponding competency questions.

As such, in this chapter we evaluate the FHN Ontology with idea of validation and verification. For the validation process, we use the competency questions that we presented in the earlier chapter, as a frame of reference to validate that the major requirements of the ontology are met. For the verification process, we use a Java EE and jQuery web-based tool that scans for major pitfalls that leads to the most common as well as uncommon modelling errors.

In the next sections we will discuss the approaches used to evaluate the FHN Ontology and conclude with a discussion on the results.

5.2. OOPS! Ontology Pitfall Scanner

OOPS! Ontology pitfall Scanner is a Java EE and jQuery web-based tool that scans for major pitfalls that leads to the most common as well as uncommon modelling errors [65].

The tool, OOPS! Ontology Pitfall Scanner has two components of verification – *diagnosis and repair*. The scanner scans the ontology upon request and automatically detects as many pitfalls as possible via the *diagnosis*, and then provides suggestions to repair the tool. The methodology used in the development of the FHN Ontology was Methontology which was

proposed by Gomez-Perez et al. [57]. Gomez-Perez was, in fact, one of the three contributors to research and development of the OOPS! Ontology Pitfall Scanner. With the backing of Gomez-Perez, the OOPS Ontology Pitfall Scanner is one of the best online tools, as it aids ontology developers in detecting ontology designing mistakes and pitfalls. The tool also expands on the list of errors detected by existing tools such as Moki, Radon and XD Analyzer [66] [78] [79].

The OOPS! Ontology Pitfall Scanner is constantly being updated. In 2013, the scanner was able to detect only 29 pitfalls [67], and now is able to detect 41 pitfalls [80]. Furthermore, the scanner also has an advanced evaluation mode that allows for evaluation classification by either *dimension* or by *evaluation criteria* [81]. This allows for an ontology designer to evaluate his ontology across in a focused dimension of concern when needed. The categorization and classification options of the OOPS! Ontology Pitfall Scanner [83], is shown below:

Evaluation Classification By Dimension:

- Structural Dimension Evaluation
- Functional Dimension Evaluation
- Usability Profiling Dimension

Evaluation Classification By Criteria:

- Consistency
- Completeness
- Conciseness

As can be seen in the categorization listed above, the OOPS! Ontology Pitfall Scanner allows flexibility while evaluating an ontology. A visualization of the structure of the two classification criteria, showing the related and overlapping pitfalls can be seen in the

images extracted from [83] and shown in **Figure 5.1 & Figure 5.2**

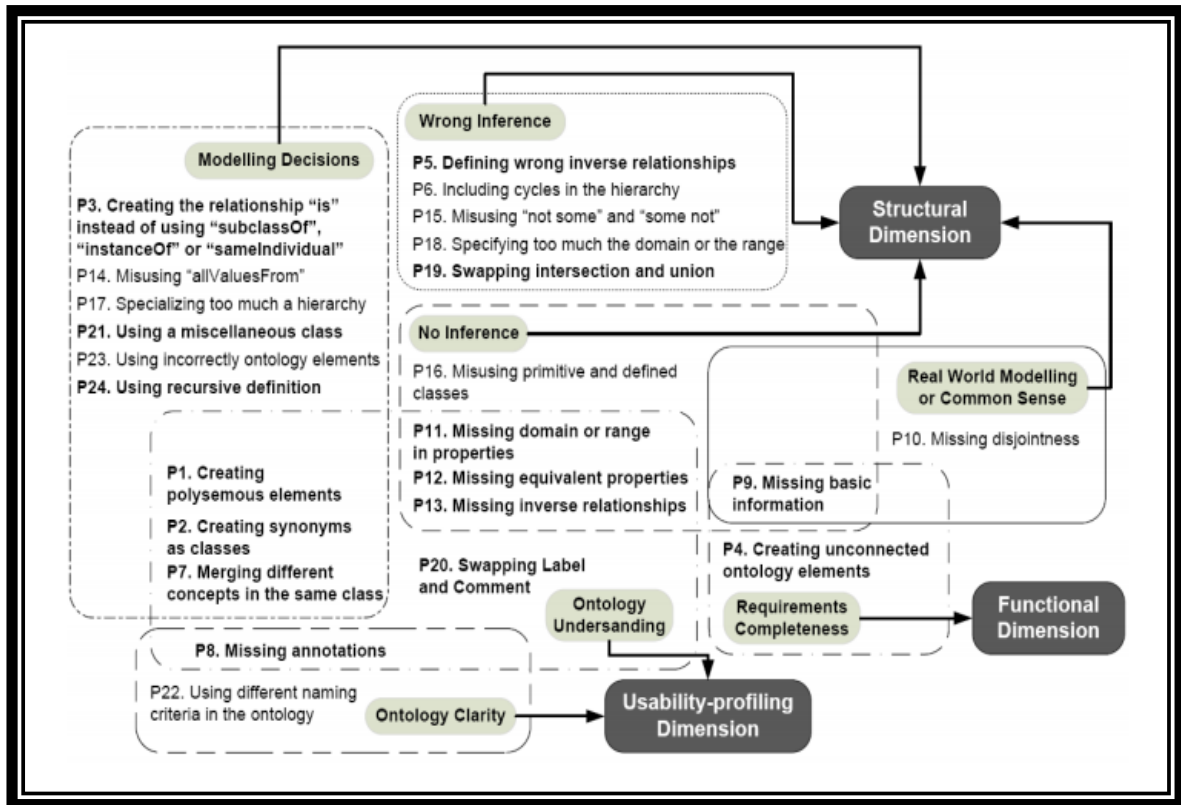


Figure 5.1: Evaluation structure of pitfalls for classification via dimensions.

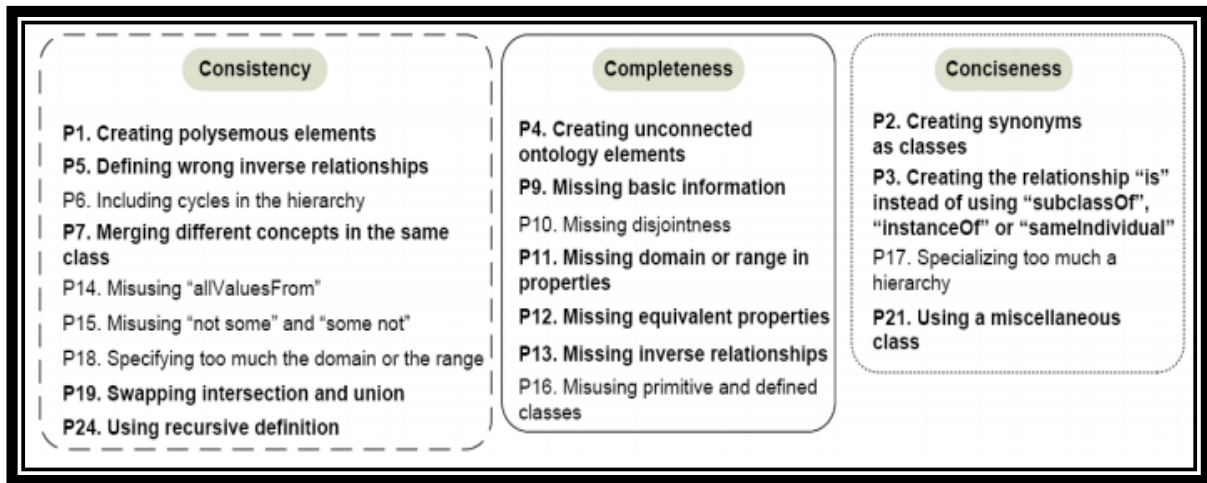


Figure 5.2: Evaluation structure of pitfalls for classification by criteria.

5.3. Evaluating Verification : Correctness & Completeness

In the previous section we discussed how fundamental workings of the OOPS! Ontology Pitfall Scanner. In this section we discuss the results of the FHN Ontology after parsing it through the scanner. Since evaluation of an ontology is an on-going, continuous process during the actual development and engineering of the ontology, there were multiple scans performed on the FHN Ontology with the OOPS! Pitfall Scanner. For illustration purposes, we show the results of the scanner when the conceptualization was just complete as well as the results of the scanner when the issues from the initial scan were addressed and fixed when possible. The results are represented on the graphs as as *"FHNOnto Pre-Cor"* and *"FHNOnto Post-Cor"* respectively.

As mentioned in the earlier chapter, the OOPS! Ontology Pitfall Scanner allows for scanning across multiple and specific dimension and criteria. As such, we ran the scanner on three well-known ontologies that are labeled as either "Good Ontologies" by the World

Wide Web Consortium [81], or deemed as an “Exemplary Ontology” by the Ontology Design Pattern Portal [82]. These ontologies are the FOAF ontology, the Marine Ontology and the Good Relations ontology. In the rest of the chapters we will provide with our results of the scanner on the three major ontologies alongside with our scanned iterations of the FHN Ontology.

5.3.1. Structural Dimension Evaluation

The graph below represents the evaluation performed according to structural dimension classification. The Structural Dimension includes pitfalls that arise from modelling decision pitfalls, incorrect inference occurrences, real-world modelling inconsistencies and bad definition structure of classes and properties.

As can be seen in the illustration, the FHN Ontology initially had a total of 33 pitfalls in this classification. After addressing all the issues produced in this classification, the next scan produced only 2 pitfalls in the structural dimension. The FOAF, Marine and Good Relations ontology all had significantly more pitfalls in this classification with each of them having over 50 pitfalls in structural dimension.

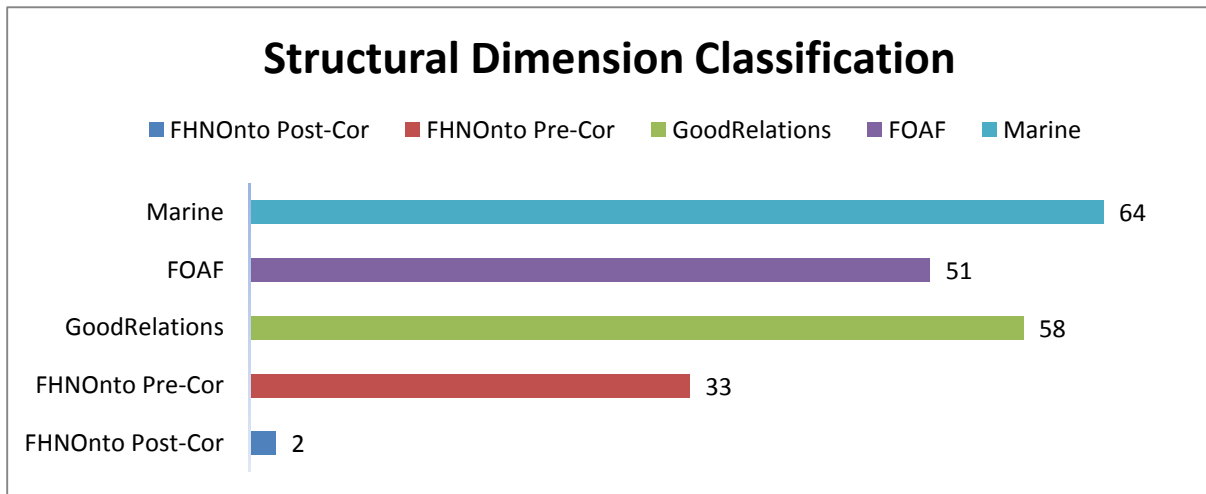


Chart I: Evaluation classified via Structural Dimension between FHN Ontology and others

5.3.2. Functional Dimension Evaluation

The next category of classification was the functional dimension. This dimension checks for pitfalls that arise from completeness of the ontology, such as having unconnected ontology classes or relationships. Often these issues can be avoided by correct designing of the relationship domain and ranges of attributes. Functional Dimension also includes pitfalls that arise from namespaces, declarations of external ontologies, missing declarations and misuse of external namespaces.

The Marine ontology and the FOAF ontology had the most amount of pitfalls at 8 each, while the Good Relations and the first iteration of the FHN Ontology scan had only two pitfalls each. After addressing the issues, the FHN Ontology was left with one pitfall that will be addressed at a later time.

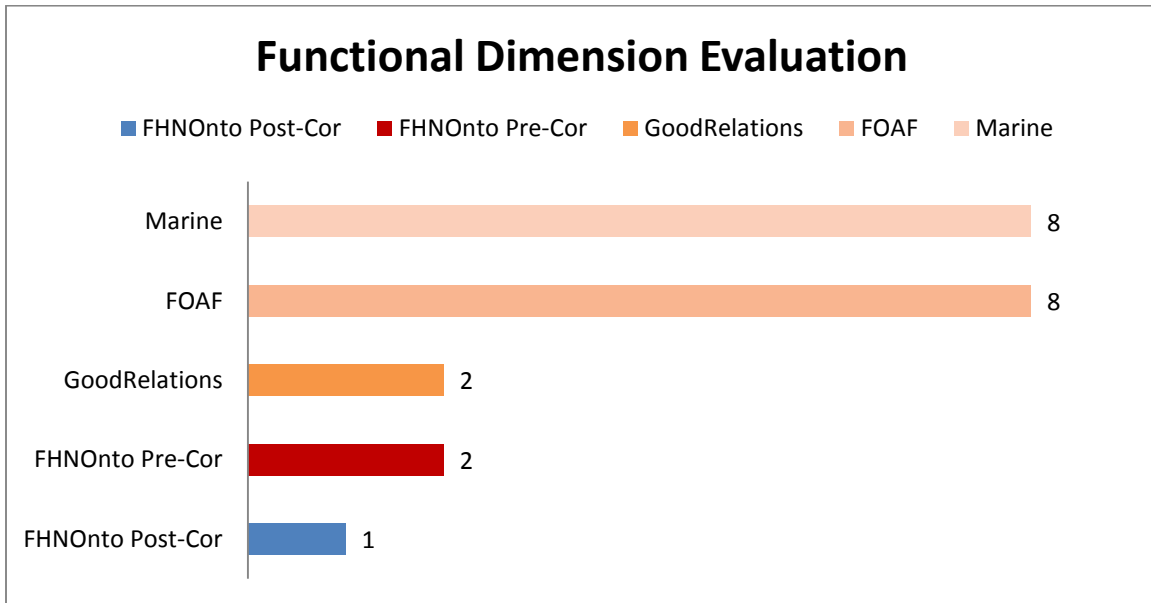


Chart II: Evaluation classified via Functional Dimension

5.3.3. Usability Profiling Dimension

The usability profiling dimension checks for pitfalls that impede the understanding, clarity and usability of the ontology from a user point of view. In this dimension, the scanner looks for missing annotations, bad class names, and misuse of annotations. Some of the pitfalls that are looked for overlap with those in the Structural Dimension, like those of missing inverse relationships and missing domain & range properties, since addition of these properties add to the overall understanding and clarity of the ontology.

As illustrated in the graph below, the FHNOnto on the first scan fared poorly in comparison to the other ontologies. The FOAF ontology fared the best, which warrants its wide usage in the semantic web.

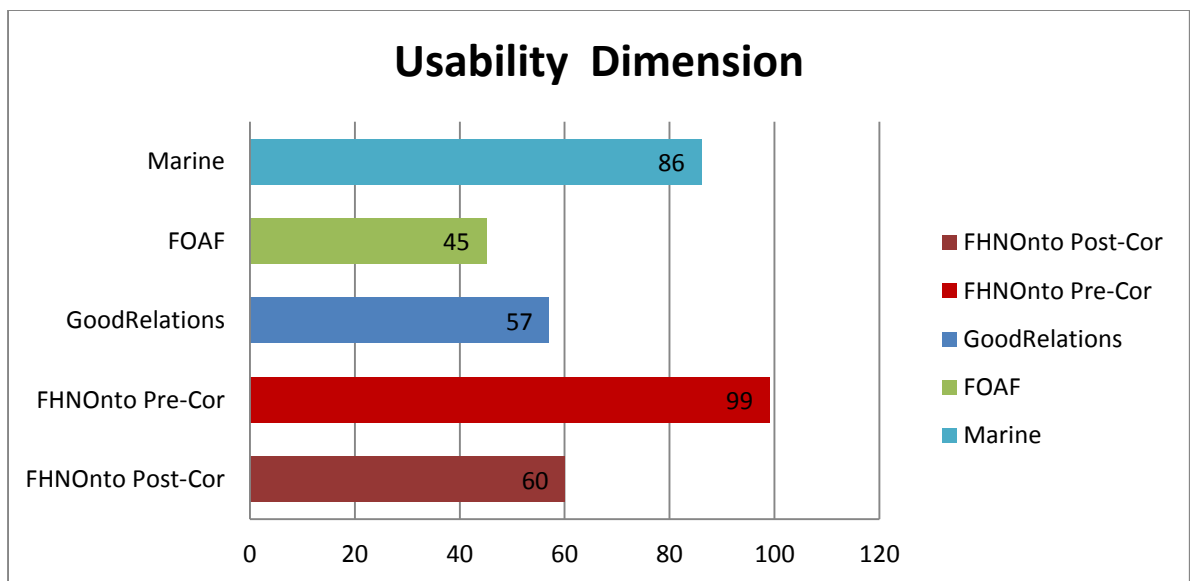


Chart III: FHN Ontology evaluation comparison by Usability Dimension

The Marine Ontology fared just as bad as the FHNOnto Pre-Cor, while the Good Relations Ontology did just as well as the FOAF ontology. After addressing some of the issues

listed by the OOPS! Ontology Pitfall Scanner, the FHNontology improved its score by 39. However, the pitfalls are still pretty high. Most of these are due to lack of annotations in the comments section

5.3.4. Consistency Criteria

This consistency criterion is one of the most important evaluation classification categories. In this criterion the scanner checks for issues that impede the consistency of the ontology. The OOPS! Ontology Pitfall Scanner will check for pitfalls that indicate issues with inverse relationships, cycles in the ontology hierarchy, and merging of concepts in the same class. Furthermore, the scanner checks for more fundamental issues like union of incompatible classes when setting domain and ranges, and using recursive definitions within the ontology.

The FOAF Ontology fared extremely well with this scan, with zero consistency pitfalls. While the FHN Ontology initially had 3 consistency issues, which were then addressed and now sits with zero consistency pitfalls as well. The Marine Ontology and the Good Relations Ontology were both scanned to have eight consistency pitfalls that could be potentially serious issues.

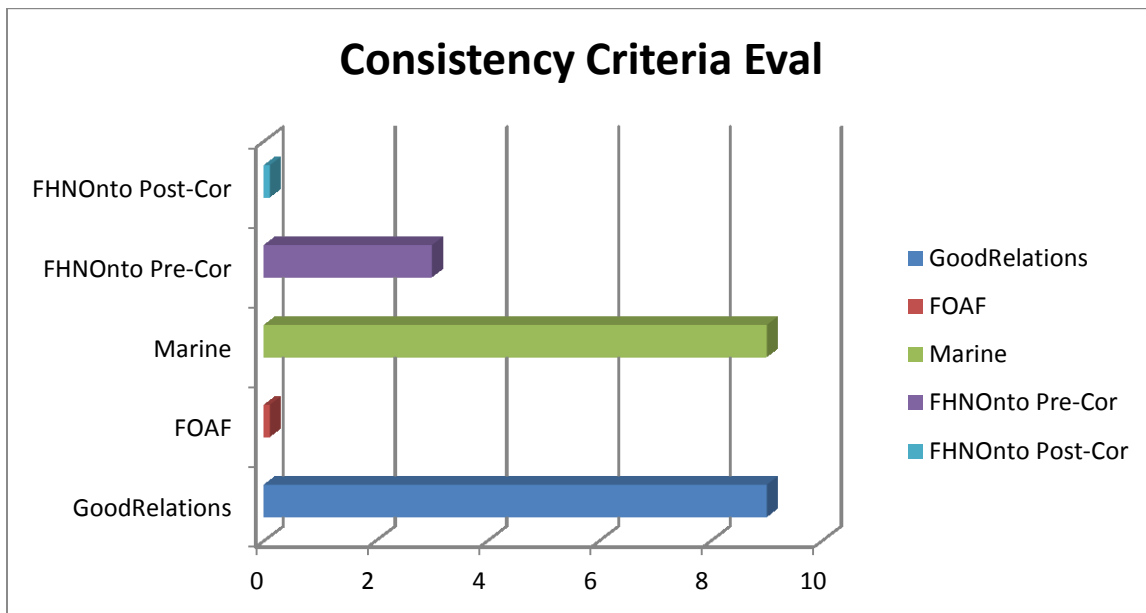


Chart IV: Criteria Evaluation comparison by Consistency

5.3.5. Conciseness Criteria

The final evaluation criteria, Conciseness, checks the ontology for pitfalls related to class and relationship design. These include pitfalls stemming from unneeded creation relationships when existing relationships within RDFs and OWL can be used, as well as using synonym classes and miscellaneous classes. These pitfalls take away clarity from the ontology and sometimes make the ontology more bulky than necessary.

The OOPS! Ontology Pitfall Scanner evaluated the FOAF, Good Relations and Marine ontologies at zero pitfalls in this criteria. The initial iteration of the FHN Ontology, however, did have a pitfall in this criteria. The pitfall stemmed from using two classes that could be set as equivalent but were not. On further examination, we also noticed that the classes were synonyms of each other. These issues were addressed, and the ontology was trimmed down to include only one entity. The ontology was scanned again, and the scanner indicated zero pitfalls in this criteria.

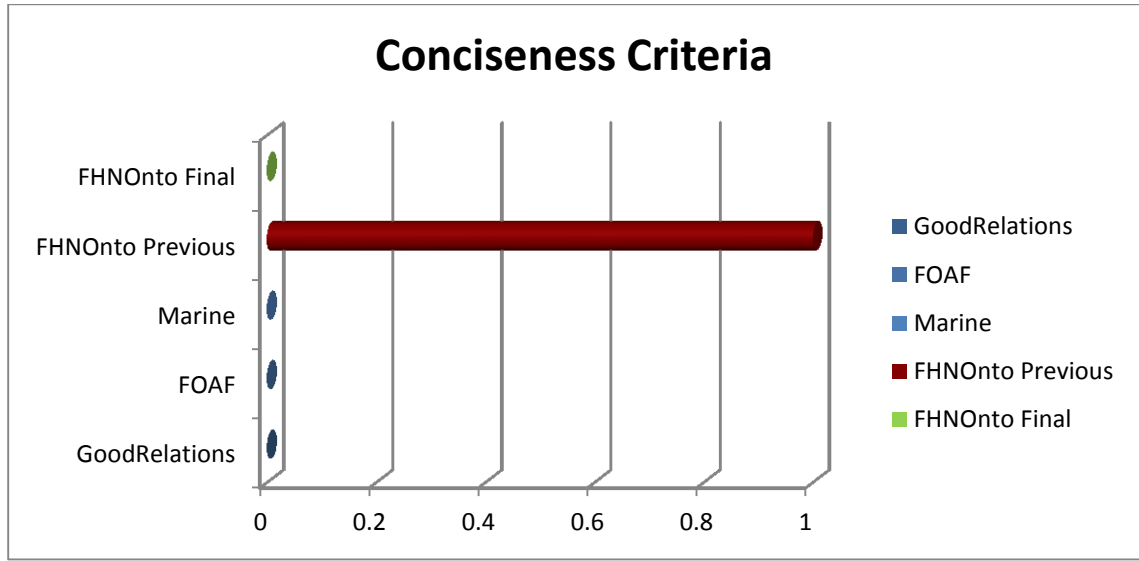


Chart V: Criteria Evaluation comparison by Conciseness

5.4. Evaluating via Competency Question

In Chapter 5, we discussed the importance of Competency Questions (CQs) in conceptualizing and engineering and ontology. CQs also play a critical role in the evaluation of an ontology after the implementation of the ontology. The CQ Doc, displayed again in **Figure 5.1**, is used as a reference to verify whether or not an ontology meets its requirements. In 2014, Yuan Ren et al. highlighted the importance competency questions and how it can be leveraged in ontology-authoring [73]. Furthermore, in 2013, Camila Bezerra et al. also stated the importance of evaluating ontologies via competency questions and proposed a mechanism to support evaluating of ontologies from their corresponding CQs [74]. In this section, we adopt a traditional and manual approach to evaluating the FHN ontology with its corresponding CW document.

In order to do so, we test the ontology by using dummy-data for the various entities. We then set out a list of expected answers for each of the questions in the CQ document. The evaluation will take place within two settings with different goals:

- A comprehensive evaluation of the FHN Ontology
- A data-flow test of the FHN Ontology System

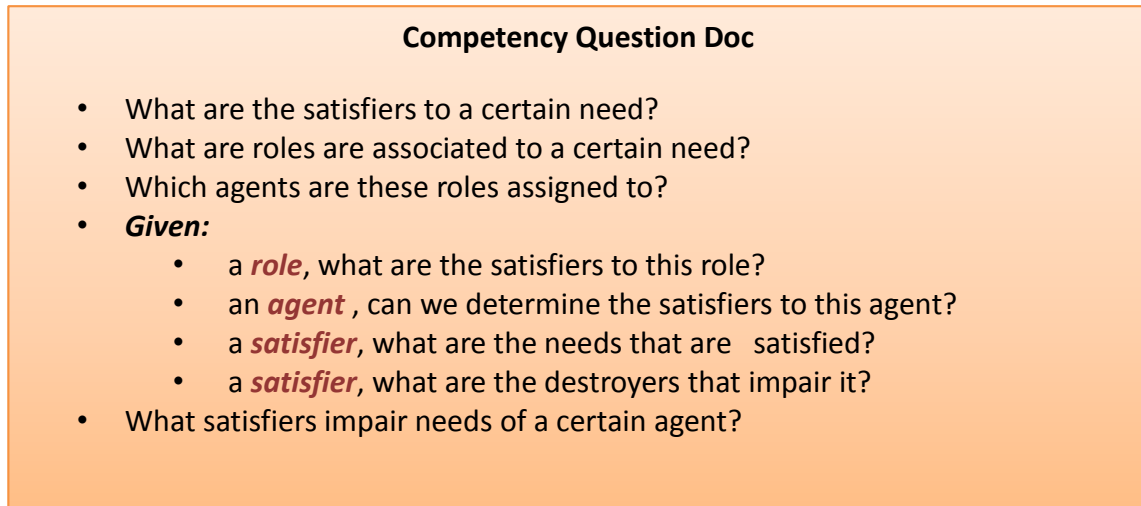
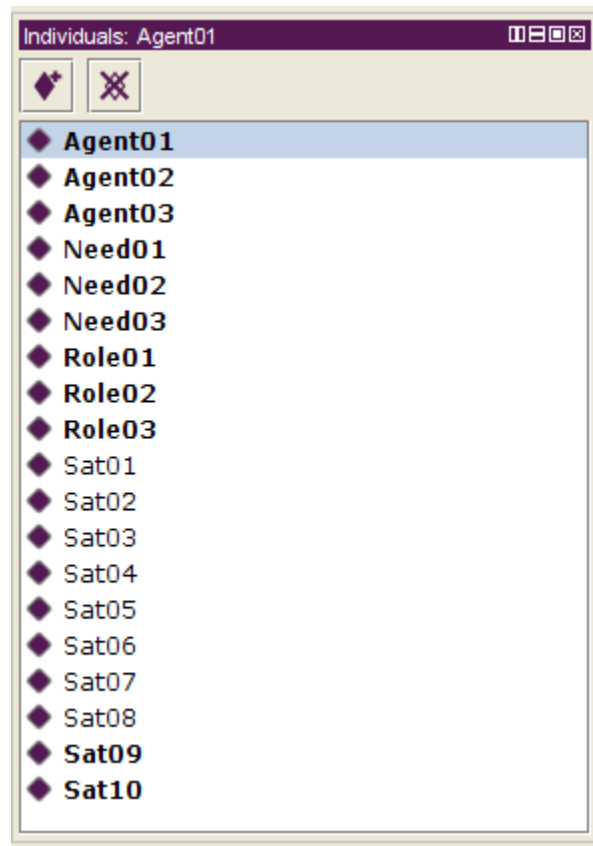


Figure 5.3: CQ Document

5.4.1. Setting Up Individuals and Asserted Facts

To set up the dummy data in a more structured and representative manner, we create a total of 19 individuals comprising of the following:

- Three Agent Individuals : Agent01, Agent02, Agent03
- Three Role Individuals : Role01, Role02, Role03
- Ten Satisfier Individuals: Sat01, Sa02 Sat10



Screenshot III: FHN Ontology Individuals for evaluation in Protégé

We then set a total of 26 assertions or facts on these 19 individuals. Some individuals have multiple facts assigned to them while others have none. The facts were set at random but were spread across the major relationships between entities. **Table 3** depicts the asserted facts, the total number of facts for the individuals that they were assigned to.

Individual	Asserted Facts	No. Of Facts
Agent01	<i>hasRole</i> Role01, <i>agentIsSatisfiedBy</i> Sat02	2
Agent02	<i>agentHasNeed</i> Need01, <i>hasRole</i> Role03, <i>hasRole</i> Role02	3
Agent03	<i>hasRole</i> Role03, <i>isRoleOf</i> Agent03	2
Role01	<i>roleHasNeed</i> Need01 , <i>roleHasNeed</i> Need02	2

Role02	<i>roleHasNeed</i> Need02,	2
Role03	<i>roleHasNeed</i> Need02, <i>roleHasNeed</i> Need03	2
Need01	<i>isSatisfiedBy</i> Sat02, <i>isSatisfiedBy</i> Sat03, <i>isSatisfiedBy</i> Sat01	3
Need02	<i>isSatisfiedBy</i> Sat04, <i>isSatisfiedBy</i> Sat05, <i>isSatisfiedBy</i> Sat06	3
Need03	<i>isSatisfiedBy</i> Sat09, <i>isSatisfiedBy</i> Sat08, <i>isANeedFor</i> Role02	3
Sat08	<i>satisfierForRole</i> Role02	1
Sat09	<i>inhibits</i> Need02	1
Sat10	<i>impairs</i> Need02, <i>satisfierForAgent</i> Agent01	2
Total		26

Table 3: Evaluation Table of Individuals and Asserted Facts.

5.4.2. Analysis of Inferred Facts on Individuals

For the next part, we run the Pellet Reasoner [91] on the ontology with the new individuals and asserted facts. Pellet is an open-source Java based OWL reasoner that performs inferences via incremental reasoning, conjunctive query answering and more [90]. The reasoner was able to establish several new facts that were not asserted directly by us. A total of 120 new facts were inferred. Since the showing all the facts would be too comprehensive and convoluted for this thesis, we split the process into two tables.

The first table, **Table 3**, depicts the total number of inferred facts for each individual, while the second table, **Table 4**, shows whether each competency question was answered or not.

Individual	Asserted Facts	Inferred Facts
Agent01	2	10
Agent02	3	12
Agent03	2	13
Role01	2	7

Role02	2	8
Role03	2	9
Need01	3	9
Need02	3	9
Need03	3	10
Sat01	0	7
Sat02	0	6
Sat03	0	9
Sat04	0	7
Sat05	0	7
Sat06	0	7
Sat07	0	0
Sat08	1	5
Sat09	1	7
Sat10	2	1
Total	26	132

Table 4: Table of Individuals & Inferred Facts.

Table 4 clearly shows that the difference in the number of facts inferred on the individuals from the asserted facts is extremely significant. **Screenshot IV and V** illustrate the inferred and asserted facts for Agent02 and Sat06. Note that Sat06 has zero asserted facts but seven inferred facts.

Description: Agent02

Types

Agent

AgentWithSatisfiers

Same Individual As

Different Individuals

Property assertions: Agent02

Object property assertions

hasRole Role01

agentHasNeed Need01

hasRole Role02

agentIsSatisfiedBy Sat09

agentIsSatisfiedBy Sat08

agentIsSatisfiedBy Sat10

agentIsSatisfiedBy Sat02

agentIsSatisfiedBy Sat01

agentIsSatisfiedBy Sat05

agentIsSatisfiedBy Sat04

agentIsSatisfiedBy Sat03

agentHasNeed Need02

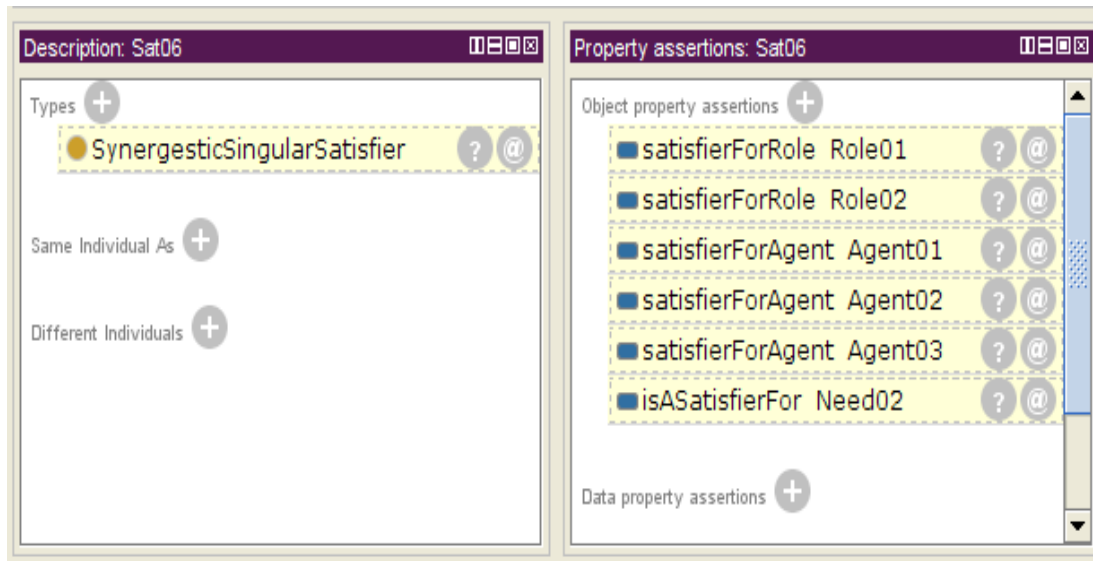
agentHasNeed Need03

Data property assertions

Reasoner active

Show Inferences

Screenshot IV: Inferred Facts (yellow highlight) and asserted facts for Agent02.



Screenshot V: Inferred Facts(yellow highlight) with no assertions for Sat06

5.4.3. Cross Referencing Inferred Facts to CQs

The final part of this evaluation was to figure out whether these inferred facts answer the questions posed in the CQ Document. **Table 5** lists the competency questions from the document in Chapter 5 and also lists whether or not the questions were answered.

Competency Question	Able to Answer?
What are the satisfiers to a certain need?	Yes
What are roles are associated to a certain need?	Yes
Which agents are these roles assigned to?	Yes
<i>Given a role</i> , what are the satisfiers to this role?	Yes

Given an agent , can we determine the satisfiers to this agent?	Yes
Given a satisfier , what are the needs that are satisfied?	Yes
Given a need , what are the destroyers that impair it?	Yes
What satisfiers impair needs of a certain agent?	Yes

Table 5: Evaluation of Competency Questions

As can be seen in the **Table 5**, the FHN Ontology was able to successfully answer the Competency Questions with the dummy-data. Even though the data consists of only 19 individuals and 26 asserted facts, the FHN Ontology shows that given a set of data, the answers can be derived. Considering that with only 26 asserted facts, the ontology inferred 120 new facts, it is certain with more data, even more answers can be derived.

5.5. Discussion on Evaluation

In section **7.3**, we evaluated the FHN Ontology for its correctness & completeness via the OOPS! Ontology Pitfall Scanner. While, in section **7.4**, we evaluated the ontology via the CQ document to validate whether the ontology answers the questions it was set out to do. Both evaluations showed favorable results. The OOPS! Ontology Pitfall Scanner was able to point out minor as well as critical pitfalls that were missed during the design of the ontology. The scanner allowed us to approach evaluation and development through an iterative manner, and thus allowing us to fix the pitfalls after detection. Graphical representation comparing the pitfalls in the FHN Ontology to major and well-known ontologies shows that the FHN Ontology

scored well in the Structural, Functional Dimension and the Consistency Criteria. However, the ontology needs improvement in the Usability Profiling Dimension due to lack of internal annotations. In the Conciseness Criteria, the FHN Ontology was the only ontology to have a pitfall indicated. However, after assessing the pitfall, it was rectified. The final FHN Ontology cleared the scan with zero pitfalls in this criteria thereafter.

In section **7.4**, we used dummy-data to show that if the ontology is given data, answers about the data can be inferred in accordance. The test-data included 19 individuals, and 26 facts, and was able to infer 132 additional facts about in total. The one hundred and thirty two facts included the answers to the questions that were posed in the CQ document as well as more complicated and extensive questions that were not addressed in the CQ document. With these results, we can safely say that the ontology has not only met the initial requirements, but has surpassed them significantly.

CHAPTER 6

Conclusion

This thesis presents the conceptualization of an OWL vocabulary that describes the *Human Needs* domain along with its relationships with the concepts of *Agents* and their *Roles*. It begins with a brief background on related technologies and concepts like the Semantic Web, the Internet of Things, Smart Cities and how these technologies are able to merge in order to share data with more meaning and context [1]. We then discuss our motivation and the problem statement, in Section 1.2 where we point out an inherent need for a machine understandable vocabulary that will allow for computers to understand human needs.

The thesis then discusses related works such as the OLA Ontology, that inferred emotions [37], and the MCRS, which was a recommender system that used existing ontologies as its base to improve their recommendations [42]. In the literature review section, we also discuss and highlight the importance of the Linking Open Data Project, that links open data sets like FreeBase and DBpedia that conform to OWL [95][96][6].

We then do a brief background discussion on the human-needs model used: the Fundamental Human Needs by Manfred Max-Neef and his colleagues, and also summarize OWL ontologies. The thesis then begins discussing the conceptualization process of the methodology used to design the ontology: *METHONTOLOGY*. We go over the major entities, properties and the implementation of the ontology and the semantic web rules (SWRL).

We further illustrate the entire cloud-based solution of FHN Ontology System by depicting the component diagram, the domain model and the sequence interaction diagram of the

system. We discuss the major components external libraries and the native android app used for prototyping before moving on to evaluation. We evaluate the ontology through two major methods, the first method used a verified ontology pitfall scanner, that allows for evaluation across multiple dimension and criteria; and the second method incorporated evaluation via the competency question formulated during the design phase. After depicting the results via tables, and bar charts, the thesis discusses the evaluation results and points out where the ontology needs to improve.

6.1. Thesis Contribution

The thesis aimed toward providing a shared and interoperable vocabulary that allows for computer systems to better understand and conceptualize human needs in an attempt to eventually provide trimmed and uncluttered recommendations as well as strengthening other ontology based systems. As it stands currently, there is no conceptualization of a vocabulary that encapsulates the human needs domain. Hence the core problem statement lies within the fact that even though there is a core need for new computing to be in more sync to human needs, there is no such vocabulary that will allow computers to understand human needs.

In this thesis we lay the foundation toward taking the first steps in developing an ontology that conceptualizes human needs. The provided vocabulary was conceptualized by using various validated fundamentals, the major fundamental being Max-Neef's interpretation of human needs being finite and consistent. We use the Max-Neef's understanding of human needs and his matrix to conceptualize human needs into an ontology along with including the concepts of an Agent and Role that complete the ontology. The ontology also sits on a repository system who's querying, updating and editing features are exposed through a RESTful web-service and thus making it accessible over the cloud for various application layers to access the ontology through the cloud as a Software as a Service (SaaS).

6.2. Concluding Statements & Experiment Results

In this thesis we successfully established a working OWL ontology modeled on the Fundamental Human Needs by Manfred Max-Neef. The goals of the FHN Ontology was corroborated by implementing a prototype android application where we extracted user data available on a social-network to successfully provide with recommendations that fulfil the user's need based on his "Role". Furthermore, we implemented a RESTful web service that holds the ontology and also connects to other linked-data knowledge bases such as Dbpedia [64] and hence contributing to the linked-data initiative.

The evaluation of the ontology showed that the FHN Ontology is free from most major, intermediate and minor pitfalls. Furthermore, the results showed that the FHN Ontology had significantly better results in the Structural, Functional Dimensional and Consistency Criteria scans when compared to other well-known ontologies. The scanner showed that the other ontologies, on average, had 28%, 9% and 6% more pitfalls in those criteria respectively. However, the FHN Ontology proved to have negligible differences in the number of pitfalls when compared to the same ontologies in the Usability and Conciseness Criteria. Future work on improving the usability of the ontology is definitely in order.

The evaluation against the CQ document showed that the ontology met its aim by successfully answering all the questions in the document with the data provided. Furthermore, the FHN Ontology was able to infer answers to questions that were not listed in the document. With the test data, the FHN Ontology was provided with only 26 stated facts. With these 26 facts, the FHN Ontology was able to infer 132 new facts. This shows that the ontology not only meets the requirements of the Competency Question document, but also exceeds it. Given more facts, the FHN Ontology can infer that much more new facts based on the properties of the facts accordingly.

6.3. Future Work

Since this undertaking is only the beginning of what is to be a lengthy and deep project, there is a lot of future work to be done. In this section we will discuss the immediate and long-term goals of the FHN Ontology and the overall system.

Further Evaluation and Specific Domain Integration: As discussed in the evaluation section, there are multiple evaluation approaches that could be used. For future work, the ontology should be evaluated as it continues growing. Future evaluation approaches include task-based, data-driven and user-based evaluation techniques [77].

A note-worthy evaluation technique that will be well-suited for an ontology such as this is a pattern-based evaluation. This will require an experience ontology-designer to look for how the ontology could use existing ontology design solutions that would fit this ontology better, if they exist.

Since the needs-domain is so vast, a specific sub-domain needs to be tackled as the first approach. Information retrieval and a domain expert will be key in this area. The ontology will then need to have a new set of CQ's that are more specific to the domain. Evaluation based on CQs will have to be performed in specific for each domain.

An example of a sub-domain and for future work, would be targeting the Travel Industry, where various human-actors have different roles according to the purpose of their role. A person travelling for the purpose of business would have different needs and satisfiers compared to a person travelling for the purpose of backpacking for adventure. As such, future work would involve specific domain integration.

Extending and Improving the FHN Ontology: There is scope for the ontology to be extended in several ways. However, the immediate and long-term goals for the FHN Ontology are to include extending the OWL to include the ability to do *arithmetic* and other advanced

calculations. This can be done via imports or using more complex axioms and would allow for the FHN Ontology to infer *satisfiers* based on their overall satisfaction value to the particular *agent* in concern.

Furthermore, an immediate goal is to extend the SWRL to **SQWRL**. SQWRL stands for Semantic Query-Enhanced Web Rule language that allows for advanced SQL-like operations in order to retrieve even more knowledge from an ontology [97]. SQWRL effectively builds a query language on top of SWRL by defining a library of SWRL built-ins. The use of SQWRL, coupled with arithmetic operations built-in, will allow the FHN Ontology to make even more powerful inferences.

End-Point & REST Services Offerings: Currently, the FHN Ontology end-point sits on an application-server that is not run constantly on a permanent basis. Future work entails hosting the end-point on a public domain with security features while also ensuring accessibility. The goal is to contribute the FHN Ontology, as it continues growing, toward the linked-data initiative and the semantic web as a whole. Furthermore as information retrieval and each specific domain conform to the FHN Ontology, the functionality and end-point offerings of the web-service should continually increase in order to allow for external clients to connect. The combination of these offerings will contribute greatly to the linked-data initiative.

References

- [1] J. Hendler et. al., "Integrating Applications on the Semantic Web," in Journal of the Institute of Electrical Engineers of Japan, Vol 122(10), October, 2002, pp. 676-680.
- [2] C. Harrison et al. "Foundations for smarter cities," in *IBM Journal of Research and Development* 54.4, 2010, pp. 1-16.
- [3] P. Barnaghi et al. "Semantics for the Internet of Things: early progress and back to the future," in *International Journal on Semantic Web and Information Systems (IJSWIS)* 8.1, 2012, pp. 1-21.
- [4] A. Fehske et al., "The global footprint of mobile communications: The ecological and economic perspective. Communications Magazine," in IEEE, 49(8), 2011, pp. 55-62.
- [5] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific American, 2001, 35-43.
- [6] C. Bizer, H. Tom, and T. Berners-Lee. "Linked data-the story so far," in *International journal on semantic web and information systems* 5.3 , 2009, pp. 1-22.
- [7] Y. L. Huahai Yang, "Identifying User Needs from Social Media," IBM Research Division, San Jose, 2013.
- [8] M. A. Max-Neef, "Human Scale Development," New York: Apex Press, 1991.
- [9] A. H. Maslow, "A theory of human motivation," in *Psychological review*, vol 50(4), pp. 370, Jan. 1943.

- [10] J. K. Ford. "Maslow's Hierarchy of Needs" in *Brands laid bare: Using market research for evidence-based brand management*, John Wiley & Sons, pp. 40, 2005.
- [11] M. A. Wahba and L. G. Bridwell, "Maslow reconsidered: A review of research on the need hierarchy theory," in *Organizational behavior and human performance*, 15(2), 1976, pp. 212-240.
- [12] L. Geller. "The Failure of Self-Actualization Theory A Critique of Carl Rogers and Abraham Maslow," in *Journal of Humanistic Psychology*, vol 22(4), 1982, pp. 56-73.
- [13] R. Cianci, P.A Gambrel, "Maslow's hierarchy of needs: Does it apply in a collectivist culture," in *Journal of Applied Management and Entrepreneurship* vol 8(2), 2003, pp. 143–161.
- [14] N. Goodwin, J. A. Nelson, F. Ackerman, & T. Weisskopf, "Consumer Behaviour: The Marketing View," in *Consumption and the consumer society*, 2008, pp. 2-3.
- [15] D. L. McGuinness, F. Van-Harmelen. "OWL web ontology language overview." *W3C recommendation* 10(10), 2004.
- [16] A. Gómez-Pérez and M. Fernández-Lopez "Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web," Springer Science & Business Media, 2006.
- [17] T. R. Gruber, "A transition approach to portable ontology specification," in *Knowledge Acquisition*, Vol. 5, 1993, pp. 199-220.
- [18] D. Gašević, D. Djurić and V. Devedžić, "Model Driven Architecture and Ontology Development," Springer, 2006.

- [19] M.A. Musen, "Automated support for building and extending expert models," *Machine Learning*, Vol. 3, 1989, pp. 347-376.
- [20] A. Gangemi. "Ontology design patterns for semantic web content," in *The Semantic Web—ISWC 2005*. Springer Berlin Heidelberg, 2005, pp. 262-276.
- [21] V. Presutti, A. Gangemi. (2008). "Content ontology design patterns as practical building blocks for web ontologies," in *Conceptual Modeling-ER*, Springer Berlin Heidelberg, 2008 pp. 128-141.
- [22] E. Miller, "An introduction to the resource description framework," in *Bulletin of the American Society for Information Science and Technology*, 25(1), 2006, pp. 15-19.
- [23] G. Klynn, J.J. Carroll, "Resource description framework (RDF): Concepts and abstract syntax," 2006.
- [24] M. Gruninger, "M.S. Fox Methodology for the Design and Evaluation of Ontologies," in *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing*, IJCAI-95, Montreal, 1995
- [25] M. O'Connor et al., "Supporting rule system interoperability on the semantic web with SWRL," in *The Semantic Web—ISWC 2005* (pp. 974-986). Springer Berlin Heidelberg, 2005
- [26] I. Horrocks et al., "SWRL: A semantic web rule language combining OWL and RuleML," in *W3C Member submission 21*, no. 79, 2004
- [27] C. Golbreich, & A. Imai, "Combining SWRL rules and OWL ontologies with Protégé OWL Plugin, Jess, and Racer," in *7th International Protégé Conference, Bethesda, MD*, July 2004
- [28] T. B. Lee et al., "The Semantic Web," *Scientific American*, 284(5), 2004, pp. 28-37.

- [29] "XML and Semantic Web W3C Standards Timeline," 2012-02-04, Available: [http://www.academia.edu/2986178/XML and Semantic Web W3C Standards Timeline](http://www.academia.edu/2986178/XML_and_Semantic_Web_W3C_Standards_Timeline) : Accessed 03/12/2015
- [30] S. Humbetov, "Data-intensive computing with map-reduce and hadoop," in *Application of Information and Communication Technologies (AICT), 2012 6th International Conference on*. IEEE, 2012.
- [31] M. Selvage, et al. (2006), "Achieve semantic interoperability in a SOA, Patterns and best practices," in IBM Technical library, Available at: <http://www.ibm.com/developerworks/webservices/library/ws-soa-seminterop/index.html>
- [32] P. Barnaghi, W. Wang, C. Henson, K. Taylor, "Semantics for the Internet of Things: early progress and back to the future," in *International Journal on Semantic Web and Information Systems*, IGI Global, September 2012.
- [33] A. R. Hevner, S.T. March, J. Park, & S. Pam, "Design Science in Information Systems Research," in *MIS Quarterly* , 28 (1) , 2004, pp: 75-105.
- [34] D. Fällman, "Design oriented-research versus Research-oriented Design," in *Workshop Paper, CHI 2004 Workshop on Design and HCI, Conference on Human Factors in Computing Systems, CHI* (pp. 24-29).
- [35] V. Eyharabide, et al., "Personalized e-learning environments: considering students' contexts," in *Proceedings of WCCE 2009, World Conference on Computers in Education Ifip - International Federation for Information Processing*, 2009
- [36] V. Eyharabide & A. Amandi, "An Ontology-Driven Conceptual Model of User Profiles," In

Proceedings of ASAI, Simposio Argentino de Inteligencia Artificial, Argentina (2007)

[37] V. Eyharabide et al. "An ontology for predicting students' emotions during a quiz. Comparison with self-reported emotions," in *Affective Computational Intelligence (WACI), 2011 IEEE Workshop on* (pp. 1-8). IEEE, 2011, pp. 1-8.

[38] G. Schreiber et al., "Semantic annotation and search of cultural-heritage collections: The MultimediaN e-culture demonstrator," in *Web Semantics: Science, Services and Agents on the World Wide Web* 6(4), pp. 243–249, 2008.

[39] E. Hyvönen et al., "Building a national semantic web ontology and ontology service infrastructure—the FinnONTO approach," In *Proceedings of the European Semantic Web Conference ESWC 2008*, Tenerife, Spain, Springer, 2008.

[40] T. Ruotsalo et al., "Smartmuseum--Personalized Context-aware Access to Digital Cultural Heritage," 2008.

[41] F. Draganidis et al., "An ontology based tool for competency management and learning paths," in *6th International Conference on Knowledge Management I-KNOW*, 2006, pp. 1-10.

[42] D. Bouneffouf et al., "Following the User's Interests in Mobile Context-Aware Recommender Systems: The Hybrid-e-greedy Algorithm," In *Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference*, IEEE, March, 2012, pp. 657-662.

[43] G. Samaras, C. Panayiotou, "Personalized portals for the wireless user based on mobile agents," in *Proc. 2nd Int'l workshop on Mobile Commerce*, pp. 70-74, 2002.

[44] V. Varma et al., Personalized web search engine for mobile devices, Int'l Workshop on

Intelligent Information Access, IIIA'06, 2006.

[45] T. Berners-Lee et al., "The Semantic Web, A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities", Science and Technology, Scientific American.com, 2001.

[46] C. Bizer et al. "Linked data on the web" in *Proceedings of the 17th international conference on World Wide Web* (pp. 1265-1266). ACM, 2008

[47] T. Berners-Lee, W3C Design Issues, Linked Data. 2006. Available:

<http://www.w3.org/DesignIssues/LinkedData.html>

[48] R. Karp et al., XOL: An XML-Based Ontology Exchange Language (version 0.4), Aug. 1999,

[49] S. Luke, J. Heflin, "SHOE (Simple HTML Ontology Extensions) 1.01 Proposed Specification", SHOE Project, Feb. 2000.

[50] N. F. Noy and D. L. McGuinness, "What is an ontology and why we need it?", Stanford University, Stanford, CA, 94305, Available:

http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html, 2008

[51] B. Chandrasekaran et al., "What are ontologies, and why do we need them?." *IEEE Intelligent systems* 14.1 1999, pp. 20-26.

[52] N. Guarin, "Formal ontology and information systems". *Formal Ontology in Information Systems: Proceedings of the First International Conference (FIOS'98), June 6-8, Trento, Italy*. IOS Press, 1998, pp. 3 ff.

[53] J. Heflin, "OWL Web Ontology Language-Use Cases and Requirements," in *W3C*

Recommendation, 10, 12, 2004

[54] N. Guarino, "Formal ontology in information systems," in *Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy* (Vol. 46). IOS press, 2008

[55] D. L. McGuinness, and F. V. Harmelen, "OWL web ontology language overview," in *W3C recommendation*, 10(10), 2004

[56] M. Dean et al., "OWL web ontology language reference." In *W3C Recommendation February 10* (2004)

[57] M.F López, et al. "Building a chemical ontology using methontology and the ontology design environment," in *IEEE intelligent Systems* 14.1 , 1999, pp. 37-46.

[58] Content Ontology Design Pattern Submission: AgentRole. Available: <http://ontologydesignpatterns.org/wiki/Submissions:AgentRole> Accessed: 06/02/2015

[59] P. Adamczyk et al. "Rest and web services: In theory and in practice." In *REST: from research to practice*. Springer New York, 2011, pp. 35-57.

[60] *The Sesame Framework*, Available: <http://rdf4j.org/about.docbook?view> Accessed: 06/02/2015

[61] *Jersey RESTful Web Services in Java*, Available: <https://jersey.java.net/#2.17> Accessed: 06/02/2015

[62] *Apache Tomcat Web Application Deployment* <https://tomcat.apache.org/tomcat-7.0/doc/deployer-howto.html> Accessed: 06/02/2015

[63] *Google API Console*: <https://code.google.com/apis/console> Accessed: 06/02/2015

- [64] *DBPedia*: <http://wiki.dbpedia.org> Accessed: 06/02/2015
- [65] M. Poveda-Villalón et al., "Validating ontologies with OOPS!" *Knowledge Engineering and Knowledge Management*. Springer Berlin Heidelberg, 2012. 267-281.
- [66] M. Poveda-Villalón et al., "Did you validate your ontology? OOPS!," (2012).
- [67] M. C. Keetl et al., "The current landscape of pitfalls in ontologies," *In Proceedings of the 5th International Conference on Knowledge Engineering and Ontology Development*, Vilamoura, Portugal, 2013
- [68] M. Fernández-López, A. Gómez-Pérez, and N. Juristo. "Methontology: from ontological art towards ontological engineering." (1997).
- [69] O. Corcho, et al. "Building legal ontologies with METHONTOLOGY and WebODE," in *Law and the semantic web*. Springer Berlin Heidelberg, 2005, pp. 142-157.
- [70] A. Gómez-Pérez, M. Fernández-López, and O. Corcho, "Ontological Engineering: with examples from the areas of knowledge management," London: Springer-Verlan, 2004.
- [71] A. Gómez-Pérez, "Some Ideas and Examples to Evaluate Ontologies," in *Technical report KSL-94-65, Knowledge System Laboratory, Stanford Univ*, 1994.
- [72] A. Gómez-Pérez and J. Pazos, "Evaluation and assessment of knowledge sharing technology," in *Mars, N.J. (ed.): "Towards Very Large Knowledge Bases," IOS Press*, 1995, pp. 289-296.
- [73] Y. Ren et al. "Towards competency question-driven ontology authoring," in *The Semantic Web: Trends and Challenges*. Springer International Publishing, 2014, pp. 752-767.

- [74] C. Bezerra et al., "Evaluating ontologies with competency questions," *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*. Vol. 3. IEEE, 2013.
- [75] M. C. Suárez-Figueroa, "NeOn Methodology for Building Ontology Networks" PhD Thesis, *Specification, Scheduling and Reuse*. Universidad Politécnica de Madrid, Spain (June 2010)
- [76] M. Poveda-Villalón and A. Gómez-Pérez, "Validating ontologies with OOPS!" In *Knowledge Engineering and Knowledge Management*. Springer Berlin Heidelberg, 2012. 267-281.
- [77] H. Hlomani, and D. Stacey, "Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey," in *Semantic Web Journal*, 2014, pp. 1-5.
- [78] *RaDON*, Available: <http://neon-toolkit.org/wiki/RaDON.html> Accessed: 06/03/2015
- [79] *XDTools*, Available: <http://neon-toolkit.org/wiki/XDTools.html> Accessed: 06/03/2015
- [80] *OOPS! Ontology Pitfall Scanner!* Available: <http://oops.linkeddata.es/catalogue.jsp> Accessed: 06/03/2015
- [81] Good Ontologies – W3C
http://www.w3.org/wiki/Good_Ontologies#The_Friend_Of_A_Friend_.28FOAF.29_ontology
Accessed: 06/03/2015
- [82] Exemplary Ontology Catalogue, Available:
<http://ontologydesignpatterns.org/wiki/Ontology:Main> Accessed: 06/03/2015

- [83] V. Poveda et al., "A double classification of common pitfalls in ontologies." (2010).
- [84] B. Shneiderman, *Leonardo's laptop: human needs and the new computing technologies*. Mit Press, 2003, pp. x – xi.
- [85] J. Tierney, "Do you suffer from decision fatigue," in *New York Times*, August 17, 2011.
- [86] Baumeister, Roy. "The psychology of irrationality: Why people make foolish, self-defeating choices," in *The psychology of economics decisions*, 1, 2003, pp. 3-16.
- [87] M. C. Suárez-Figueroa et al., "Ontology Engineering in a Networked World," Springer, Berlin, 2012.
- [88] Keshk Mohamed and Sally Chambless. Model Driven Ontology: A New Methodology for Ontology Development. November 2008.
- [89] Antonio De Nicola, Michele Missikoff, and Roberto Navigli. A software engineering approach to ontology building. *Information Systems*, 34:258–275, April 2009.
- [90] Pellet – W3C Semantic Web, Available: <https://www.w3.org/2001/sw/wiki/Pellet> Accessed: 09/03/2015
- [91] Pellet Home Page – Complexible, Available: <http://pellet.owdl.com/> Accessed: 09/03/2015
- [92] M. Sabou and M. Fernandez, "Ontology (Network) Evaluation," in *Ontology engineering in a networked world*, Springer Berlin Heidelberg, 2012, pp. 193-212.
- [93] Cyganiak, Richard, and Anja Jentzsch. "Linking open data cloud diagram." *LOD Community*

(<http://lod-cloud.net/>) (2011).

[94] S. Auer, et al. "*Dbpedia: A nucleus for a web of open data*," in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Volume 4825, 2007, pp 722-735

[95] Meij, Edgar, et al. "Mapping queries to the Linking Open Data cloud: A case study using DBpedia," in *Web Semantics: Science, Services and Agents on the World Wide Web 9.4* (2011): 418-433.

[96] Linked Data – Home - <http://linkeddata.org/> Accessed: 11/03/2015

[97] O'Connor, Martin J., and Amar K. Das. "SQWRL: A Query Language for OWL," in *OWLED*. Vol. 529. 2009.

[98] B. Robert, et al. "The million dollar programming prize," in *IEEE Spectrum* 5.46, 2009, pp. 28-33.

[99] I. Horrocks et al. "Semantic web architecture: Stack or two towers?" in *Principles and practice of semantic web reasoning*. Springer Berlin Heidelberg, 2005, pp: 37-41.