A Dissertation On

ON-TO-METHODOLOGY:

A GENERIC ONTOLOGY DEVELOPMENT

Submitted in the partial fulfillment of the requirements

For the award of the degree of

Master of Technology in

SOFTWARE ENGINEERING

By

ANUPRIYA TEWARI

(2K12/SWE/08)

Under the Esteemed guidance of

PROF. DAYA GUPTA

DEPARTMENT OF COMPUTER ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

BAWANA ROAD, DELHI-110042

CERTIFICATE

Delhi Technological University
(Government of Delhi NCR)
Bawana Road, New Delhi-42

This is to certify that the thesis entitled **"On-To-Methodology: A Generic Ontology Development"** submitted by **ANUPRIYA TEWARI** (Roll Number: **2K12/SWE/08**) for the partial fulfillment of the requirements for the award of degree of **Master of Technology** in **Software Engineering** in the **Department of Computer Engineering**, Delhi Technological University, New Delhi is an authentic work carried out by her under my guidance.

Project Guide:
Prof. Daya Gupta
Professor
Department of Computer Engineering
Delhi Technological University, Delhi

ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude and respect towards my guide **Prof. Daya Gupta, Department of Computer Engineering.**

I am very much indebted to her for the generosity, expertise and guidance I have received from her while working on this project. Without her support and timely guidance the completion of the project would have seemed a far –fetched dream. In this respect I find myself lucky to have her as my guide. She have guided not only with the subject matter, but also taught the proper style and techniques of documentation and presentation. I would also like to take this opportunity to present my sincere regards to **Ms. Shruti Jaiswal**, Research Scholar, DTU for extending her support and valuable Guidance.

Besides my guides, I would like to thank entire teaching and non-teaching staff in the Department of Computer Engineering, DTU for all their help during my tenure at DTU. Kudos to all my friends at DTU for thought provoking discussions and making the stay very pleasant.

Anupriya Tewari
M.Tech
Software Engineering
2K12/SWE/08

ABSTRACT

Ontologies are used for representation and reuse of domain knowledge. The way how ontologies are created is very important. The process of ontology development influences the usability of the ontology outside its primary application and its suitability for integration with other ontologies. A few research groups are now proposing a series of steps and methodologies for developing ontologies. However, mainly due to the fact that Ontological Engineering is still a relatively immature discipline, each work group employs its own methodology. This brings inconsistency of viewpoints into consideration because different methodologies are best suitable for different domains along with the issue that various stakeholders are associated with a particular project and all stakeholders have a different view of the project.

Hence new approaches to ontology creation are necessary to bridge the "ontology gap" and enable domain experts to create formalized knowledge. In this work, I will be proposing an ontology development methodology called *On-To-Methodology* that is comprehensive & generic. I elicit the domain knowledge in a structured manner and employ various documents to support the development process.

A case study on the healthcare domain is also performed for illustration of the methodology. An evaluation has been performed to check the consistency, correctness and completeness of our methodology. A comparison of the methodology has been made with the popular ontology development methodology named “METHONTOLOGY”, which helps in evaluating the effectiveness of the proposed method. A tool titled *On-To-Methodology Tool* has been created to support the development and evaluation of *On-To-Methodology*.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: INTRODUCTION

This chapter gives introduction to ontology and presents the motivation, scope and problem statement of the project. The chapter ends with a brief description of how this thesis is organised.

1.1 General Concepts

This section introduces to the basic introductory concepts of ontology.

1.1.1. What is Ontology?

The origin of the word "ontology" is rooted in the Latin word *ontologia* (ont + logia) [1]. It was introduced first in 19th century as a philosophy discipline. In computer science, ontologies were introduced for knowledge sharing [2] and reuse. Very often the word 'ontology' is considered as a branch of philosophy discipline that is the science of the kinds and structures of objects, events, properties, processes and relations in every area of reality [3]. It is also considered as a branch of 'metaphysics' that deals with the nature of being.

However, there is no universally accepted definition to define ontology, but most precisely it can be defined as: "*Ontology is a formal, explicit specification of a shared conceptualisation*" [4, 5]. In Computer Science it can be described as a hierarchical data structure that contains all the relevant entities (related to any real world domain), their properties, the relationships among them and rules of inference within a domain.

1.1.2. Why is Ontology Used?

In recent years, ontologies have become very popular. Huge study material is available on the World Wide Web concerning ontologies. The selected reasons for why one should use/develop ontology are following [6]:

i. Sharing a common understanding of the organization of information among different software or different people.
ii. Enabling reuse of knowledge related to any domain.
iii. Differentiating the domain knowledge from functional knowledge.
iv. Analyzing the domain knowledge.
v. Making clear domain assumptions.

It is now a well known fact that constructing a domain model, or ontology, is a very important step in the development of recent knowledge based systems. The advantages of ontologies have been widely recognized, and they include enabling of sharing and re use of knowledge and better engineering of knowledge based systems with respect to its acquisition, verification and maintenance.

Recently, ontologies are also being used as centrally controlled dictionaries that are integrated into different catalogues, databases, web applications, bodies of knowledge, knowledge management applications etc.

1.1.3. Application Areas of Ontology

The main application areas of ontology are following:

i. *Artificial intelligence*: Ontologies are being used in various applications of AI such as, Natural Language processing, knowledge management etc.
ii. *Semantic Web*: Ontologies are used to add semantics (meaning) to knowledge.
iii. *Software engineering:* Ontologies can be used for different software engineering activities.

iv. *Biomedical informatics*: Popular ontologies are available in bioinformatics, for example: Gene Ontology [8].

v. *Library science*: Ontologies for conceptualizing knowledge of library domain are also available.

vi. Enterprise bookmarking

vii. Information architecture

viii. *Security requirements engineering*: Ontology can also be used for security requirements analysis. They provide the knowledge base for completeness and consistency checking of the elicited requirements. The advantage of using ontology for requirements analysis is that it allows the requirements engineers to analyze the requirements with respect to the semantics of the domain. We proposed a framework for security requirements analysis using ontology in a previous study [7].

1.1.4. Types of Ontology

Ontologies have been categorised in various categories, by different authors based on different criteria. Some categories [9] are as follows:

i. High-level Ontology

The ontologies that fall under this category explain common ideas, such as space, time, materials and items. They are not constrained to a particular domain or problem. Their objective is to unify requirements between a large group of users. The *Gene Ontology* [8] comes under this category.

ii. Domain Ontology

Domain ontology models knowledge related to a specific domain, and models a constituent of the world. Domain ontology provides the meanings of terms applied to that domain. For example the word *card* has different meanings. An ontology in the domain of *poker* would define the *playing card* meaning of the word, while an ontology for the domain of *computer hardware* would define the *graphics card* meaning. *OntoDesign* [10] can be taken as an example of this category.

iii. Upper Ontology

Similar to high level ontology this category is a model of all the general objects, and is applicable across a wide range of domain ontologies. *Bunge-Wand-Weber (BWW)* [11] is perhaps the most widely used ontology in Information Systems.

iv. Application Ontology

These ontologies explain concepts that belong at the same time to a domain and a process, by using the ideas of both, the domain ontologies and task ontologies. *Experimental Factor Ontology (EFO)* [12] is an example of this category.

v. Generic Ontology

These are also called common sense ontologies. These ontologies present knowledge that spans various domains and provide common information of the real world. They provide concepts and ideas for area, time, state, activities, etc., and model knowledge about common or specific entities. As an example, *Generic Ontology Matching and Mapping Management (GOMMO)* [9] can be considered.

vi. Representational Ontology

The ontologies that fall under this category are not attached to any particular domain. They provide information in terms of concepts, without developing what they might signify. Hence, they determine notions, which show information in an object or framework focused approach. *Ontology Based Knowledge Representation (OBKR)* [13] can be taken as an example.

vii. Terminological Ontology

The ontologies in this category model and explain the expressions to be used in a particular problem domain. They can also be used to acquire a new language for solving a particular problem. *Universal Medical Language System (UMLS)* [14] is an example of terminological ontologies.

viii. Information ontology

This category consists of ontologies that deal with consistent storage of information. For example, *Information Artifact Ontology (lAO)* [9].

ix. Knowledge Representation Ontology

The ontologies in this category specify knowledge conceptualizations with an internal structure. A particular knowledge use is focused upon and described by these ontologies.

x. Static Ontology

These ontologies consider the static features, connections and relationships of the entities they describe. They consider that some features of the objects in the world do not change. As an example, we consider *Resource Ontology*.

xi. Dynamic Ontology

In contrary to the abovementioned static ontology this category describes those ontologies that consider patterns for knowledge representation. They consider that there are patterns that keep changing in the world. To design this kind of ontology it may be necessary to use limited state machines, Petri nets, etc. 'Process', 'condition', or 'stat conversions' are some of the main concepts in these [9].

xii. Social Ontology

These ontologies are focused on explaining the social factors such as, an organization culture, the hierarchy, the interdependencies etc. thus, the main concepts used are ‘position’, ‘responsibility’, ‘authority’, ‘actor’ etc.

Other types of ontology can be intentional ontology, task ontology, method ontology etc.

1.1.5. Ontology Components

An ontology has following building blocks that define it completely:

i. *Individuals*: instances or objects are called individuals.
ii. *Classes*: Classes refer to collection of elements, concepts, programming classes, sets of elements, types of objects etc.
iii. *Attributes*: Attributes are defined as properties or features or characteristics of classes and objects.
iv. *Relations*: Relations represent different types of connections between classes and individuals.
v. *Function terms*: Complex structures formed from certain relations that can be used in place of an individual term in a statement are called functions.
vi. *Restrictions*: Constraints are formal statements of what must be true for some assertion to be accepted as input.
vii. *Rules*: Simple statements following if-then-else structure are called rules.
viii. *Axioms*: Statements (including rules) arranged in a logical format that encompass the overall knowledge that the ontology models in its application domain are called axioms.

ix. *Events*: Any changing of relations or properties is called event.

1.1.6. Ontology Development Languages

Different languages have been introduced for the development of ontologies. Following classification of ontology languages has been provided:

1.1.6.1. Logical languages: This category can be further sub divided into following sub categories:

i. First order predicate logic [15]
ii. Rule based logic
iii. Description logic

1.1.6.2. Frame based languages: These languages are similar to relational databases. Some examples are:
i. F-Logic (Frame Logic) [16]
ii. OKBC (Open Knowledge Base Connectivity) [17]
iii. KM programming Language

1.1.6.3. Graph based languages: Some examples of graph based languages are following:
i. Semantic networks

1.1.6.4. Markup Ontology Languages

The following markup languages are used for ontology construction:

i. DAML (DARPA Agent Markup Language)

It is an agent markup language based on the Resource Description Framework. DAML+OIL [18] was the updated extension of DARPA. This language used RDF and XML for a basis. RDF namespaces were used to assist in the integration of randomly incompatible ontologies. This program was initiated in 1999 by Defense Advanced Research Projects Agency (DARPA). The major contribution was given by James Hendler.

ii. Resource Description Framework (RDF)

RDF [19] is a member of the World Wide Web Consortium (W3C) specifications. It was originally designed as a metadata model. RDF is now commonly being used as method for modelling and conceptualizing the knowledge implemented in the web based resources using various notations and data serialization formats. It is widely being used in knowledge engineering.

The common serialization formats used are: JASON-LD, Turtle, N-Triples, N-Quads etc. A group of RDF statements represent a labelled, directed multi-graph. It makes statements about the resources in triplets, that is, subject- predicate-object format. This methodology provides the basis for the World Wide Web Consortium's Semantic Web platform.

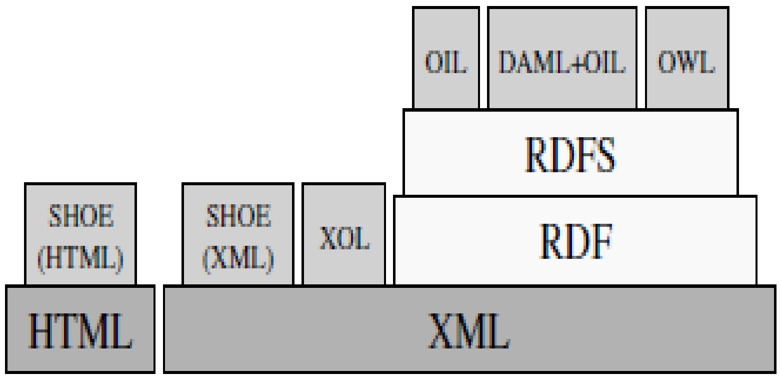
iii. Web Ontology Language (OWL)

OWL [20] is the most academically and commercially popular ontology language used for knowledge representation. It is endorsed by World Wide Web Consortium (W3C). Another version of OWL is available for use, named OWL 2. Three variants of OWL are available with different levels of expressiveness namely, OWL Lite, OWL DL and OWL Full.

iv. RDF-Schema (RDFS)

RDFS [21] is a group of classes with definite properties using the RDF extensible knowledge representation language, which provides basic elements for the explanation of ontologies, with the intent of storing the knowledge of RDF resources [6]. Figure 1 shows the stack of different ontology markup languages.

Fig 1. Ontology Markup Languages [22]



1.2. Motivation

Initial application of ontologies in computer science was as a means of providing semantics for the semantic web. Since then the applications of ontology have increased enormously. According to the increasing applications various ontology development methodologies have been proposed. But the ontologies created by different creators are different and inconsistent. There is a huge difference between different domains and currently there is no general methodology for constructing ontologies. And in spite of much being written in literature about ontologies, the number and quality of actual, ontologies available on the Web today is remarkably low [23,24].

When, we study ontologies, we discover significant variety in them, even when they have been constructed for almost similar purposes [25].This situation creates problems in ontological engineering and it must be changed. Due to this a proper understanding of how to construct ontology is required.

The recent research work on ontology development methodologies shows that there is *no unified methodology for ontology development which is capable of being used in different domains*. Different projects use different ontology development methodologies according to their domain and requirements. Thus a new generic development methodology is needed. Some other facts that motivated this research are as following:

- None of the methodologies are fully established if we compare them with the recent standards. Here we used the *Ontometric* set of ontology characteristics as the standard to evaluate an ontology and to further evaluate its construction methodology [26].

- METHONTOLOGY is the most mature development methodology still it is not provided with proper validation and maintenance. *This motivated us to extend this work and build up a new methodology that provides proper validation and maintenance.*

- Most methodologies are not defined in appropriate detail and the domain of their use is also limited to business.

- Many methodologies do not follow a life cycle. This motivated us to develop a methodology which follows a life cycle.

From above discussion it is clear that ontology development is still more of a matter of skill rather than being a well developed engineering process. Tremendous effort and biased decisions from the ontology developers are required by most of today's methodologies to construct the ontology. These methods still suffer from inconsistencies and need a new approach for construction. This motivates research in this area of ontological engineering. The future growth of ontological engineering highly depends on developing a generic methodology for ontology construction and enhancing the scope from a few projects to many available.

1.3. Problem Statement

Ontologies have become popular for knowledge representation and sharing. But our research shows that different processes that exist for ontology construction have various inconsistencies such as: domain knowledge inconsistency due to different representation of the content of the domain, dissimilarity due to difference in relationship between different concepts or language or vocabulary used to represent the concepts. There can be subjective inconsistency also because of creators' interests, taste, ability, preferences etc. All these processes lack automation which makes ontology construction, maintenance and evaluation a tedious task. Therefore, some kind of automation would guide the developing team in ontology construction, maintenance and evaluation.

However, proposal of ontology development from Methontology [29] proposes to build ontology from scratch and resolve problem of consistency by generating different documents. Also very recently a generic methodology On-To-Methodology [50] was proposed to address the above mentioned issues. Also, in this proposal an attempt was made to automate design. But these proposals could construct ontologies from scratch and their process was sequential in nature. Therefore, there is a need for an ontology process which is incremental and can modify or integrate an existing ontology to construct a new ontology. This thesis will enhance the proposed framework of On-To-Methodology by incorporating these factors and also, add a mechanism for evaluating an ontology using the technique of OntoMetric. Therefore, problem of the thesis can be stated as:

Enhance the framework for generic methodology: On-To-Methodology by making it evolutionary and add formal evaluation process that is automated.

1.4. Scope of Work

The On-To-Methodology framework [50] has 4 steps namely: Knowledge Acquisition, Design and Formalization and Maintenance. In this work, Knowledge

Acquisition is done not only from subjective domain but also from existing ontologies. During the design phase we use evolutionary design technique. Any concept can be dynamically added into the ontology location map using Jason string format. In the formalization phase we have tried to establish the generality by developing an ontology for healthcare system. The maintenance phase has also been updated by using a different insertion deletion algorithm. Finally a tool has been developed based on this framework.

An evaluation process which generates OntoMetric values for ontology has also been developed and is applicable to this framework also. Therefore scope of the work can be summarized as:

- Improve framework for ontology development: On-To-Methodology to incorporate evolutionary development.
- Adapt the different steps, named, knowledge acquisition, design, maintenance and evaluation to support incremental property and improve automation.
- To develop a tool based on this new framework that automates the design including the integration from other ontology and that embed Protégé in a user friendly manner for implementing the ontology.
- Apply the framework to construct ontology for the *hospital healthcare domain.*
- To establish the quality compare the hospital ontology created by On-To-Methodology and METHONTOLOGY using the criteria of OntoMetric.

1.5. Thesis Organisation

The remaining sections of the thesis are organised as follows:

Chapter 2 provides a detailed description about different ontology development methodologies. It provides an insight of the drawbacks of the available approaches.

Chapter 3 presents the research background of this research work.

Chapter 4 presents the proposed On-To-Methodology in detail.

Chapter 5 describes the implementation part of this research work. It describes how Hospital Ontology and the On-To-Methodology Tool are developed.

Chapter 6 shows the evaluation of both the Hospital Ontology and the On-To-Methodology approach. It also describes in detail the automation generated for evaluation.

Chapter 7 concludes the thesis and presents the possible improvements in this research work in future.

Chapter 8 exhibits the publications from this research.

Chapter 2: LITERATURE REVIEW

In this chapter we perform a small literature survey on the existing literature about different ontology development methodologies proposed by researchers in the past few years. A threefold description has been provided for each methodology, including the basic theory about the method, the steps involved and analysis of the methodology.

This chapter is divided into two sections, the first section exhibits the different methodologies for ontology development and the second section presents a comparative analysis of the surveyed studies.

2.1 METHONTOLOGY

METHONTOLOGY [28] is an ontology construction methodology that is based upon software engineering and knowledge engineering. This methodology was developed by the Laboratory of Artificial Intelligence at the University of Madrid. It enables the construction of ontologies from the smallest level of granularity that is from scratch. METHONTOLOGY is divided into different activities. The steps involved in ontology development are as follows:

2.1.1. Project management activities: These activities can be subdivided into following:

i. **Planning**: involves deciding the schedule and making other plans.
ii. **Control:** involves checking whether above decided plans are working or not.
iii. **Quality Assurance:** finally checking the quality of the ontology.

2.1.2. Development-Oriented Activities: These activities can be subdivided into following:

i. **Specification**

The goal of the specification phase is to define the ontology scope, viewpoints and to ultimately generate a formal, semi-formal or informal ontology specification document. This document can be written in natural language, using a set of intermediate representations or using competency questions.

ii. Conceptualization

In this activity, the domain vocabulary identified in the specification phase is used to define the problem domain and to organize the domain knowledge in the form of a conceptual model. Firstly, a Glossary of Terms (GT) is identified corresponding to the domain. The Glossary of Terms is a collection of all the relevant terms and their meanings that describe the domain knowledge.

Secondly, all the related concepts (nouns) and related relationships/ properties (verbs) are identified and a concepts classification tree and a verbs diagram is generated. Next the ontology development process can be disintegrated into two different, but related teams.

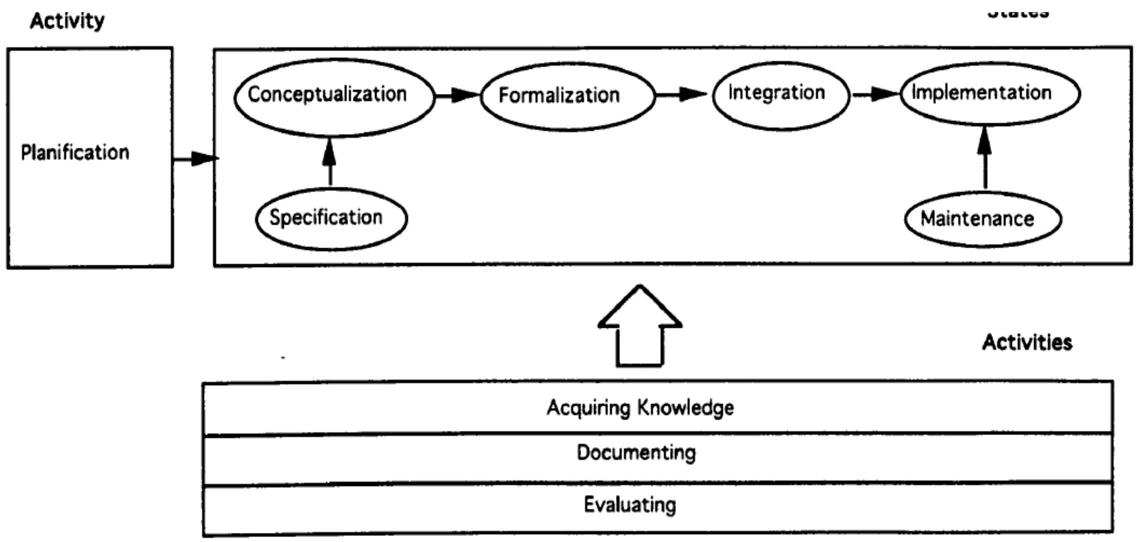
iii. Formalization

This step involves implementing the ontology in a semi formal format.

iv. Implementation

To make use of ontologies in various domains it is required to codify them and to associate them with a supporting environment. The desired output of this phase is the code of the ontology in a formal language such as: OWL, RDFS, LOOM, Ontolingua, C++ or in any formal language selected by the developer.

Fig.2: The conceptualization phase of METHONTOLOGY [28]



2.1.3. Support Activities: These activities can be subdivided into following:

i. Knowledge Acquisition

It is simultaneously performed with the other activities. Generally most of it is completed with the requirements specification phase, and the amount of acquisition decreases with the ontology development process.

ii. Evaluation

Analogous to the software engineering process, the term evaluation means involvement of verification and validation. Evaluation means carrying out a scientific judgment of the ontology. Along with the code the documentation is also evaluated with respect to a frame of reference. This frame of reference can be the specification document generated in previous phases.

The desired output of this phase is a set of evaluation documents which describe the process of ontology evaluation, the tools and techniques used, the list and kind of errors discovered at different phases, and the sources of knowledge used for the evaluation.

iii. Documentation

No standard guidelines have been set on how to document ontologies. In most of the cases, the only documentation available is in the code of the ontology in a formal language, the natural language text attached to formal definitions, and the papers published.

In METHONTOLOGY documentation is considered as a very important part of development. Thus it is involved in every phase of the ontology development process. More specifically, after the specification phase, a requirements specification document is created; after the knowledge acquisition phase, a knowledge acquisition document; after the conceptualization phase, a conceptual model document is provided; after the formalization phase, a formalization document; after the integration, an integration document; after the implementation/coding phase, the implementation document; and after the evaluation, an evaluation document.

2.1.4. Analysis of the methodology: The following results are discovered after analysis of the methodology:

- Details of steps properly provided.
- METHONTOLOGY gives freedom of choice with regard to formalization.
- Application independent strategy is used thus it can be used for different applications.
- Uses hybrid approach for concept identification.
- Recommended lifecycle is Evolving prototypes.
- Environmental pollutants ontologies, CHEMICALS ontology [29], The Reference-Ontology etc. are built using this methodology.

2.2 Methodology by Farooq et. al.

In their paper [31], they proposed a methodology for ontology design during the semantic web engineering process. The existing web application design methods may easily be upgraded for semantic web (SW) applications by incorporating their technique. They made some deliberate efforts in design phase of ontology development which they found missing in other methods.

2.2.1 Adaptations made at Specification Level

The main adaptation proposed was that along with the requirement specification performed at the specification phase a preliminary web-ontology model should also be generated. The following activities are to be followed:

i. Declaration of the Domain Vocabulary
ii. Identification of resources and group assignment
iii. Identification of Axioms
iv. Identification of relationships and name assignment
v. Identification of data-characteristics and name assignment
vi. Application of constraints
vii. Verification

2.2.2 Adaptations made at Design Level

Design phase mainly deals with transformation of the specifications received from the previous phase into an algorithmic or pseudo code format so that it can be easily coded in any programming language and become executable. Because ontology (both schema and document) is based on Resource Description Framework model, a new model was defined which was named the so called RDF model. This model was designed using the preliminary ontology model created in the previous phase. This model consisted of triples. A triple contains following three components: (i) a subject (ii) a predicate and (iii) an object.

2.2.3. Analysis of the methodology: Following are the results of analysis:

- This methodology is semi application-dependant.
- The life cycle followed for ontology construction was not clearly defined.
- The main emphasis is given on the design phase.

2.3. MADRE

The authors, Zhang et. al., realized that because of the huge diversity among the domains, no technical path was available as the only standard and unique method for construction of ontology. Thus in their work [32], they discovered and described an improved method for ontology construction called MADRE (Method of Analysis, Design, Representation and Evaluation). This method was based on IDEF5 [33] language format and followed a seven step methodology.

The key feature of this method was the use of graphic language to explicitly define domain knowledge corresponding to the research domain. This method also evaluates the ontology construction process.

The construction majorly follows four steps namely domain analysis, ontology analysis and design, ontology representation and ontology evaluation. The set of steps to be followed to construct ontology using MADRE are:

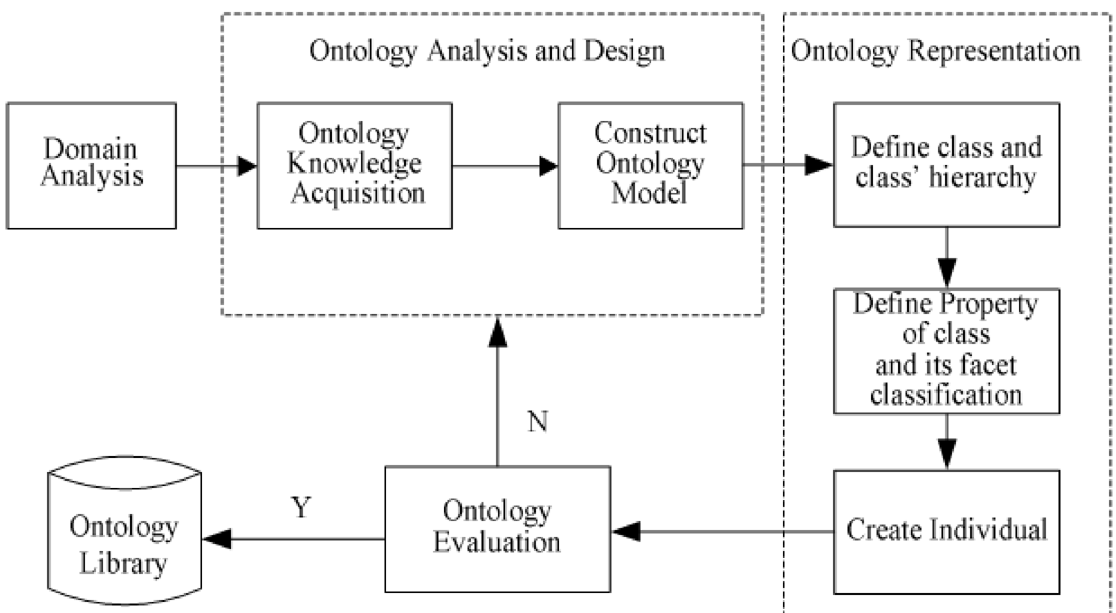
i. *Domain Analysis Phase*: This phase deals with determining the scope and extension of the domain and reusability of the ontology.

ii. *Ontology Analysis and Design Phase*: This phase deals with acquiring semantic information of the major concepts, properties, relations etc and to define the hierarchy of concepts and relations. The raw data collected in the previous phases is formally defined in this phase, together with this, the using the basic knowledge of ontologies the ontology model is generated.

iii. *Ontology Representation Phase*: This phase makes use of the ontology model developed before. In this phase, the classes, properties and individuals are identified and defined. The definition of the aforesaid is done using structural ontology language OWL.

iv. *Ontology Evaluation Phase*: This phase defines the criteria of verifying the ontology by explicitly displaying a rule set. Further evaluation is made for checking the accurateness of ontology using First-Order Logic inference. Consistent results show a successful ontology development else, Ontology Analysis and Design phase is repeated.

Fig.3: The process of MADRE ontology construction



2.3.1. Analysis of the methodology: Following are the results of analysis:

- This methodology is based on the Evolving Prototype model.
- A library of commonly used relations is used which includes definitions of different kinds of relations.
- This methodology is application independent but is used to assist in ontology construction.

2.4. Methodology by Uschold And King

In this study a methodology for ontology construction is provided by the authors. This work has its roots in the *Enterprise Ontology*. Enterprise Ontology was built for enterprise modelling processes [34]. Following steps are used by this methodology:

2.4.1. Identify purpose

Clarify the motive or purpose for building the ontology and define its intended users.

2.4.2. Ontology Capture

- Deals with identification of key concepts and relationships of the domain.
- Precisely defining the concepts and relations.
- Identification of terms to refer to the above concepts.

2.4.3. Coding

This step involves the implementation of the ontology in a formal ontology implementation language such as RDF, OWL, DAML+OIL etc.

2.4.4. Integrating existing ontologies

A developed ontology is integrated with other available open source ontologies for capturing more knowledge.

2.4.5. Evaluation

Evaluation means carrying out a scientific judgment of the ontology, with respect to its structure, documentation etc, keeping a standard frame of reference.

2.4.6. Documentation

The documentation phase is same as in the other methodologies, only the name and type of documents produced are different. This phase deals with generating documents for supporting the ontology development process.

2.4.7. Analysis of the methodology: Following are the results of analysis:

- Detailed description of techniques and activities not provided.
- No recommendations provided for formalization of knowledge.
- This technique uses an application independent approach
- Hybrid approach is used for identification of concepts.
- No life cycle proposed by the methodology.
- This methodology has been used to develop complex projects in the business domain.

2.5. Methodology by Grüninger And Fox

The authors provide a new methodology for ontology construction which is based on the TOVE project ontology. The TOVE project was developed for business processes and activities modelling [35, 36, 37]. The steps involved are as follows:

i. *Capturing of motivating scenarios*: this involves capturing of those incidents that motivated the development of the ontology.
ii. *Informal Competency Questions*: The competency questions are used to verify the ontological commitments that have been made previously to check whether the ontology meets its requirements or not.
iii. *Getting informal terminology*: The terminology is gained from the competency questions.
iv. *Getting formal terminology*: These are gathered from the informal terminology.
v. Get formal competency questions.
vi. Using formal language to describe axioms and definitions for the terms in the ontology.

vii. Establishing standard criteria to check the completeness of answers to competency questions.

2.5.8. Analysis of the methodology: Following are the results of analysis:

- Detailed description not provided for both the activities and the techniques.
- Identification of scenarios performed in the specification stage, so it follows an application-semi dependent strategy.
- The methodology does not explicitly describes a life cycle.
- Hybrid approach is used for identification of concepts.
- Complex projects in the same domain have been developed using this approach.

2.6. Methodology given by Amaya Berneras et al.

This work was done under the *KACTUS* [38] project. The goal of the KACTUS project was to check the possibility and amount of knowledge reuse in complex technical systems. The project also focused on the ontological support to knowledge reuse [39]. This methodology is limited to be applied to the development of applications only. So, each time an application is created, the ontology that represents the knowledge required for the application is also created. The steps involved are as follows:

2.6.1. Specification of the application

Analogous to the other methods this step provides the terms/concepts relevant to the application along with the components that the application tries to model.

2.6.2. Preliminary design generation

The preliminary design is developed using the top-level ontological classes**.** This process involves search of ontologies for extension of the given application.

2.6.3. Ontology refinement and structuring

This is the maintenance phase where ontology is checked for coupling and cohesion. Minimum coupling is required along with maximum cohesion.

2.6.4. Analysis of the methodology: Following are the results of analysis:

- Very little details of the methodology are provided.
- No recommendations for knowledge formalization are made.
- This method follows an application-dependent strategy
- Top down strategy for identifying concepts has been used by the authors.
- This method does not explicitly defines a life cycle, but assumes that it should be same as is used in the development of the application associated.
- Projects in the domain of electrical networks have been worked upon using this method.

2.7. The Methodology in SENSUS Approach

The SENSUS [40] ontology was developed for natural language processing at the ISI (Information Sciences Institute) natural language group to provide abroad-based conceptual structure for development of machine translators. It consists of more than 50,000 concepts organized in a hierarchy. Whenever ontology is to be developed for a domain, the SENSUS terms are appended with domain terms and irrelevant terms are pruned. The steps involved are as following:

i. A group of terms are taken as seed terms.
ii. These terms are linked to SENSUS manually.
iii. All the concepts that fall in the path starting from the seed terms to the root of SENSUS are also included.
iv. Any new terms discovered later are added.
v. If some nodes are present which witness huge number of paths through them, the entire subtree under those nodes is added. But generally adding a whole subtree is avoided.

2.7.1. Analysis of the methodology: Following are the results of analysis:

- This method is not very detailed.
- Semantic networks are recommended by this methodology.

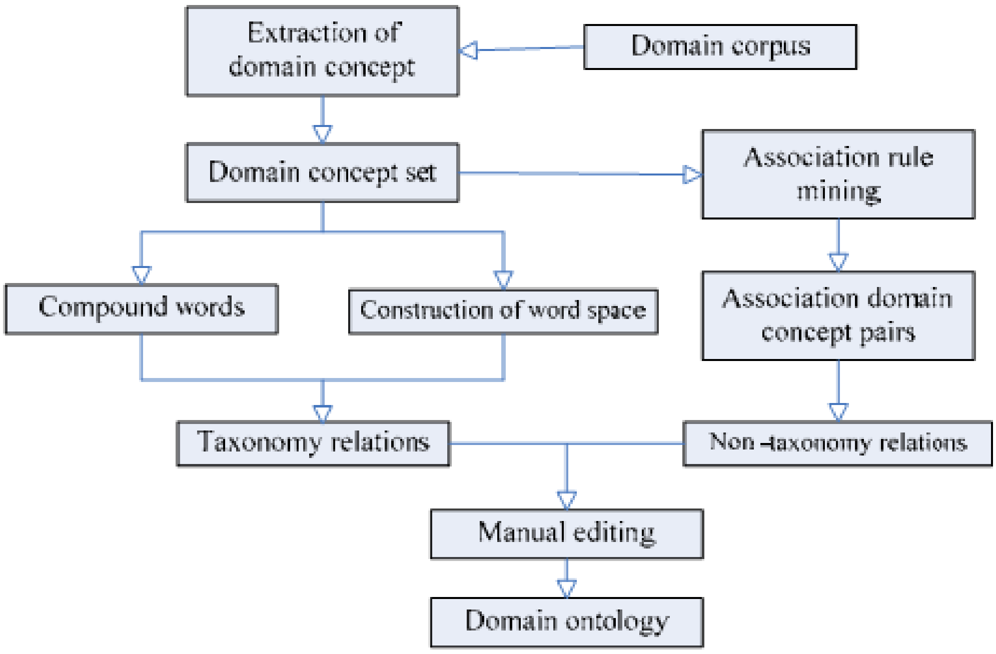
- This method is semi application independent as the seed terms come from application domain only.
- It uses a bottom up strategy by starting with the most specific concepts.
- No particular life cycle is referenced.
- This methodology was used for developing ontologies for military air campaign planning. These ontologies contained concepts such as the campaign plans, participants, scenarios etc.

2.8. Semi-automatic Domain Ontology Construction methodology by Dan et. al.

This study by Dan et. al. [41] provides semi-automatic construction of domain ontology based on Chinese word partition and data mining.

The approach in their paper mainly consists of three parts: firstly, the extraction module of domain concepts and secondly, the extraction module of taxonomic and non-taxonomic relations. Statistical analysis method, clustering, association rule mining and generalized suffix tree method are respectively used for the aforesaid sections. The system framework proposed in the research is shown in Figure 5.

Fig.4: The system framework



2.9. Comparative analysis of all surveyed methodologies

Table 1 briefly shows the comparative analysis of some of the main ontology development methodologies.

Methodology	Inheritance from knowledge engineering	Detail in descrip-tion	Domain of Applicati-on	Recomme-nded Life cycle	Dependency on Application	Design Strategy	Automa-tion Provided
Uschold And King	Partial inheritance	Not detailed	Business Enterprise	None	Independent	Hybrid	Enterpris-e Toolset tool
Grüninger And Fox	Little inheritance	Partially detailed	Virtual Enterprise domain	None	Semi de-pendent	Hybrid	None
Berneras et.al.	Huge inheritance	Not detailed	Electrical networks	None	Dependent	Top down	None
METHONTO-LOGY	Huge inheritance	Detailed	Domain of Chemicals	Evolving prototype	Independent	Hybrid	WebODE , ODE tools
SENSUS	None	Partially detailed	General Base Onology	None	Semi dependent	None	None

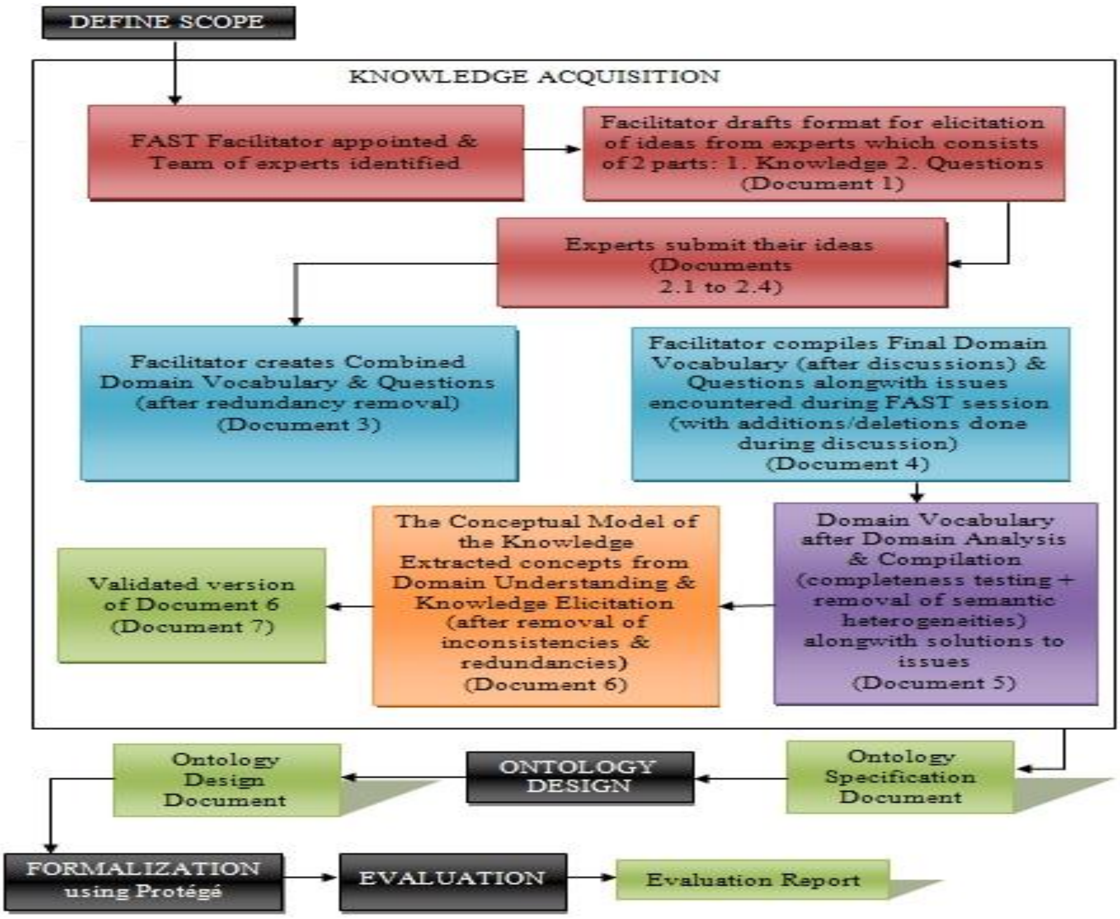
Table 1. Comparative Analysis

Chapter 3: RESEARCH BACKGROUND

This chapter gives a brief introduction to the background work used for this research. The methodology proposed in this research gets its name "On-To-Methodology" from a proposal previously proposed in [50].

In the previous approach the author proposed a methodology for ontology development which includes the following activities: Defining Scope, Knowledge Acquisition, Design, Formalization, Evaluation and Maintenance. The knowledge acquisition phase has been given major emphasis and a linear structure for construction was used in this approach. It also focuses on production of varios documents and automation of design. Figure 5 shows the various steps described in this methodology.

Fig 5. Methodology given in [50]

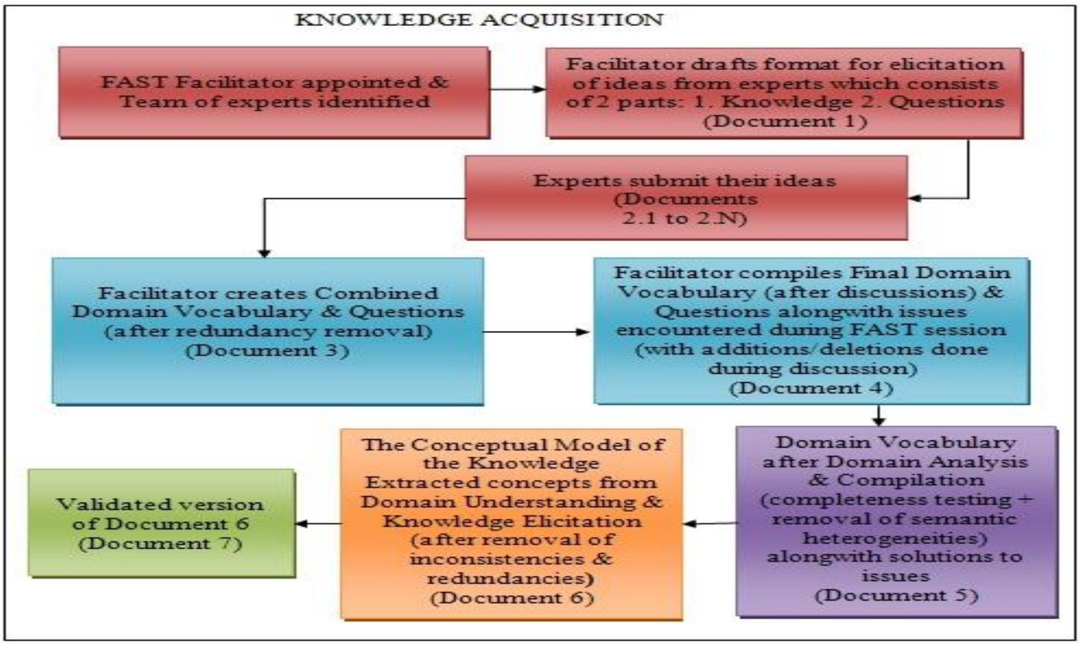


The ontology development phases described in [50] are as follows:

i. **Define Scope**: This step deals with defining the purpose and intended end users of the ontology. This step is performed by experts and users manually.

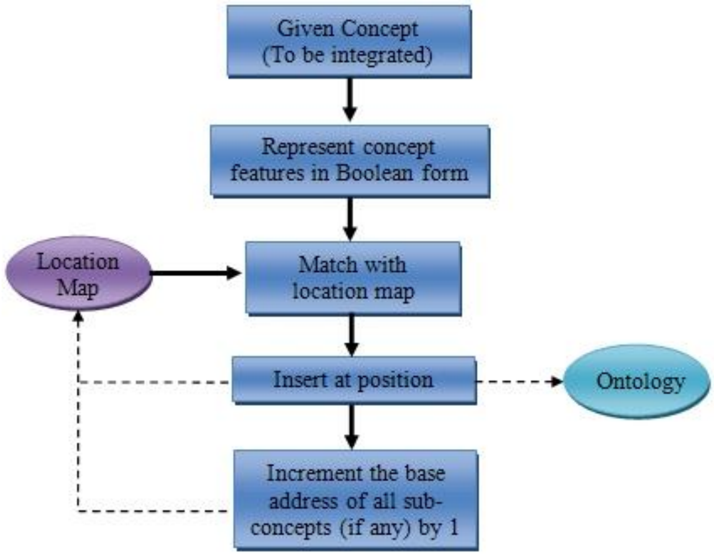
ii. **Knowledge Acquisition**: This step deals with acquiring the knowledge required for ontology construction from various resources. Only non-ontological resources such as, books, websites, experts etc. have been used for this purpose. The Facilitated Application Specification Technique (FAST) of software engineering has been used in this approach for knowledge elicitation. This step consists of following subparts: Domain Understanding & Knowledge Elicitation, Domain Analysis & Compilation, Building Conceptual Model of the Knowledge and Validation. A separate document is generated after the execution of each of these subparts. The output of this phase is the Ontology Specification Document (OSD), which can be viewed in detail in [50]. Figure 6 shows the different phases of FAST knowledge Acquisition used in this method.

Fig 6. FAST knowledge acquisition in [50]



iii. **Design Ontology**: This step forms the structure of the ontology using the OSD and location map. The framework used for ontology design is given in figure 7. The output of this phase is the Ontology Design Document (ODD), which can be viewed in detail in [50].

Fig 7. The design framework in [50]



The algorithm used for design is as follows:

i. Given an Ontology structure with a concept *Thing* which is super concept of all concepts.

ii. A concept with from Conceptual model of the knowledge is selected. Let it be X.

iii. All features for that concept are identified and labeled with a feature number as follows:

{X} → f^+ <feature no. from 1 to n>.

iv. The concept is then represented in the Boolean form as follows:

C (X): $\{f^+ 1 \bullet f^+ 2 \bullet \ldots \bullet f^+ n\}$

v. The concept represented in Boolean form is compared to other concepts in the location map one by one. Say the concept currently chosen from the Ontology structure be Y with address in location map as (a, b). Following cases can happen:

a) If {Y}: $\{f^+ 1 \bullet f^+ 2 \bullet \ldots \bullet f^+ (n-k)\}$ where $k<n$, and we have

{X}: $\{f^+ 1 \bullet f^+ 2 \bullet \ldots \bullet f^+ (n-k) \bullet \ldots \bullet f^+ n\}$

In this case there is a match between X and Y for f^+1 to $f^+ (n-k)$. This means that X is a child of Y with $f^+ (n-k+1)$ to $f^+ n$ as additional features. Thus X is inserted in the Location map with address (a+1, xx).

b) If {Y}: $\{f^+ 1 \bullet f^+ 2 \bullet \ldots \bullet f^+ m\}$, and we have

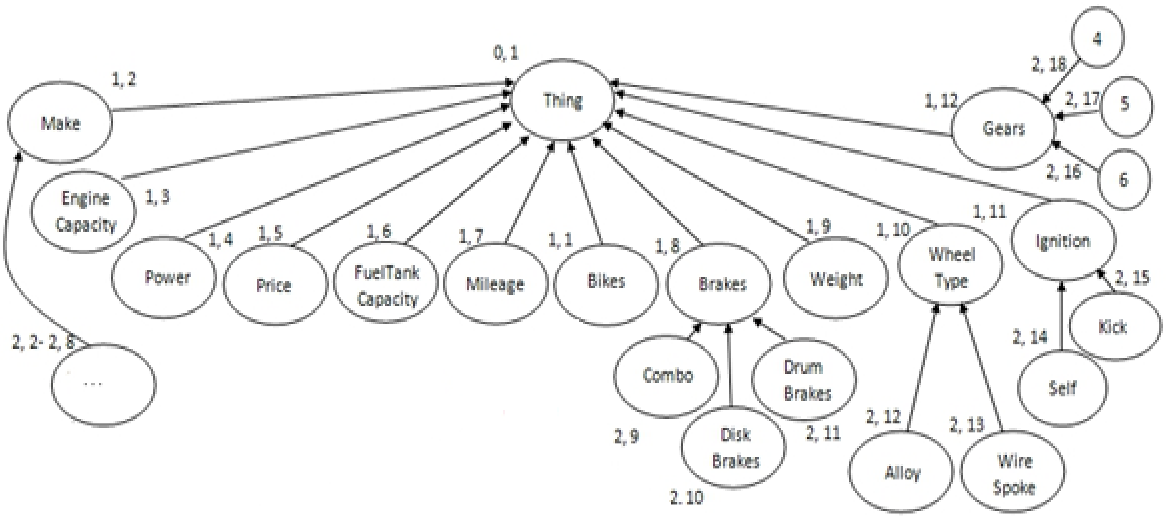
$\{X\}: \{f^{+}1 \bullet f^{+}2 \bullet \ldots \bullet f^{+}n\}$

If there exists a feature in Y that does not exists in X, then X is not a sub concept of Y, thus it is either a brother concept of Y or sub concept of any other concept.

vi. Steps ii to v are repeated for all concepts in the conceptual model and the end result is the complete ontology structure.

iv. **Formalization:** This phase deals with implementing the ontology. OWL was used for ontology implementation. The output of this phase is the developed ontology. Figure 8 shows the structure of ontology built in [50].

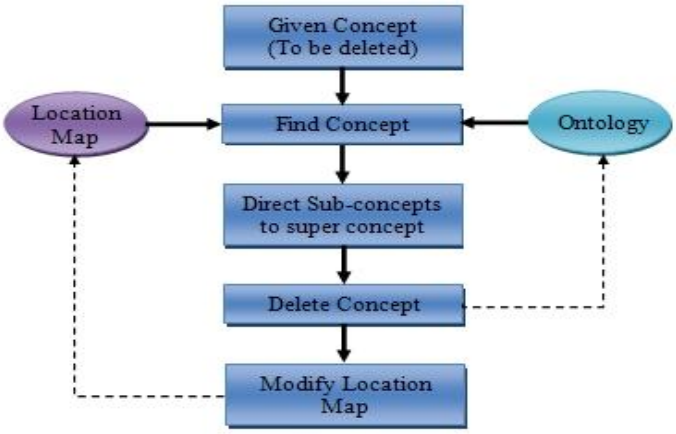
Fig 8. Ontology structure of ontology for bikes



v. **Evaluation:** After the ontology development it is evaluated for correctness in relations and correctness in hierarchy. The output of this phase is the evaluation report which provides only semi automatic evaluation of the ontology using OWLViz. The output can be seen in detail in [50].

vi. **Maintenance:** This step deals with insertion and deletion of concepts in the ontology. The framework for insertion is same as given in figure 6. The framework for deletion is given in figure 9.

Fig 9. The deletion framework in [50]



Advantages of the methodology: The methodology in [50] had following advantages:

i. This approach helps in developing a better understanding of the domain by applying group oriented discussions, such as, FAST, for knowledge elicitation.
ii. The methodology attempts to solve the problem of viewpoint inconsistency by performing validation at the end of knowledge acquisition phase and using group based activities.
iii. Verification and validation activities are embedded at early stages in the methodology for prevention and early detection of faults in the ontology structure.
iv. The methodology provides automation for ontology design.
v. Extensive documentation is performed in this approach for making the development process more formal.
vi. Maintenance is also considered as part of ontology construction which helps in keeping the ontology updated.

Drawbacks of the methodology: The methodology in [50] had following advantages:

i. The knowledge acquisition approach is not generic. The methodology uses only FAST as its standard knowledge acquisition technique, which might not be optimum in various cases. It also makes use of only non ontological resources for elicitation of knowledge and does not elicits knowledge form other ontologies.
ii. This methodology uses a small ontology of bikes with less than 50 concepts, for its illustration, which results into a weak case study. Moreover, the automation developed for its support automates only the design phase of development.
iii. No particular life cycle is described, but in practice, the methodology follows a sequential model for development.
iv. The evaluation phase is highly overlooked in this approach. Also, this phase is completely manual and gives only theoretic results in the form of the Evaluation report.
v. The approaches used for maintenance lack in completeness. For example, the same framework is used for deletion of a single concept and for deletion of a sub ontology.

Chapter 4: PROPOSED METHODOLOGY FOR ONTOLOGY DEVELOPMENT

This chapter describes our proposed framework for ontology construction. This process is generic and it is an extended and improved version of proposed methodology in [50]. This methodology is analogous to the software engineering process. Hence, different documents are produced after each phase of development for better understanding and record keeping. The methodology follows the *iterative and incremental development approach*. The methodology will be illustrated through a case study in further chapters. Moreover, we also propose an automated tool and evaluation approach for the proposed methodology in chapters 4 and 5 respectively.

4.1. The Proposed Framework: Enhanced On-To-Methodology

The different phases in the Enhanced On-To-Methodology are described in Figure 10. These consist of five main activities namely Knowledge Acquisition, Design Ontology, Formalization, Implementation and Maintenance. The output of Knowledge Acquisition is Ontology Specification document. The output of Design Ontology is Ontology Design Document (ODD) and consists of logical design of the ontology being developed. The output of implementation is the developed ontology. The output of Maintenance will be the modified ontology (if any modifications are made). Along with these five main activities some parallel activities are also involved in each phase of development. These are, parallel knowledge acquisition, integration, documentation, evaluation and configuration management.

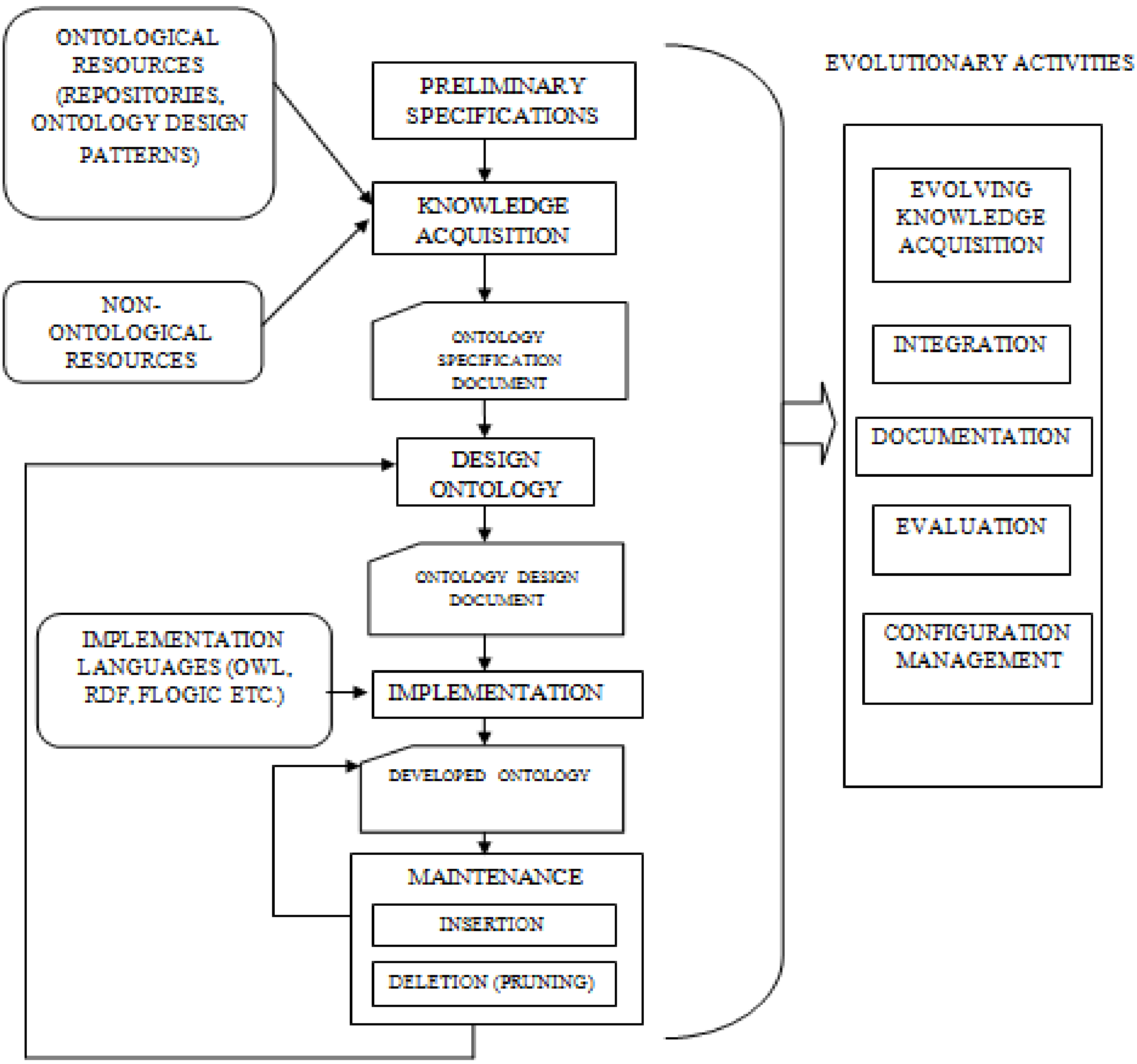
4.1.1. Preliminary Specifications

This phase is the beginning point of ontology construction. It deals with defining the scope of the ontology. This activity consists of identifying and defining the purpose of the ontology, its viewpoints and usage scenarios. Identification of viewpoints is emphasized in this step as different stakeholders may define different requirements for the ontology.

4.1.2. Knowledge Acquisition

This activity deals with acquiring knowledge from various resources, as knowledge is the raw material for ontology construction. The domain knowledge can be gained from ontological resources such as whole ontology, ontology modules and/or ontology statements, as well as non ontological resources such as:

Fig 10. On-To-Methodology: Ontology Development Methodology



- Dictionaries
- Taxonomies
- Books
- Experts
- Tables
- Files
- Web Pages
- Other Open Source Ontologies
- Bill of materials (BOMS)

We can use any standard old methods for elicitation of knowledge such as:

- Brainstorming
- FAST(Facilitated Application Specification Technique)
- Peer Interviews
- Expert Interviews etc.

We chose FAST to resolve the viewpoints inconsistency, as it is a formal group oriented technique. The output of this phase is an Ontology Specification Document. This phase can be further divided into following sub phases:

4.1.2.1 Domain Understanding & Knowledge Elicitation

In this step, the rough knowledge is identified and elicited from different sources in the form of individual terms. During collection of terms for the vocabulary the elements of ontology are kept in mind for better elicitation. The elements of ontology are: Class, Relationships, Constraints, Forms, Instances, Constants, and Instance attributes. These are explained in [37] as follows:

- *Classes/Concepts*: Represent a particular set of similar objects in the domain.
- *Properties*: Represent features of concepts.

- *Relationships*: Represent a property between two classes such as 'isSubClassOf' relation.
- *Constraints*: These are the conditions of domain and range which must be defined and satisfied.
- *Instance*: Particular values assigned to a class objects.

These elements can be elicited using any systematic elicitation method such as using competency questions, machine learning tools etc, depending upon the viewpoints described in the above step. Here we don't categorize the domain keywords according to the above categories but the focus is on comprehensive list of constituents without worrying about redundancies or overlaps.

Competency questions are also elicited in this phase, by the domain experts. These questions are queries related to the ontology which deal with the intended use of the ontology. The competency questions are later used to evaluate the ontology.

4.1.2.2. Analysis of knowledge and conflict resolution

This step deals with manual analysis of the ontology. Analyzing the elicited knowledge at this early stage helps in better ontology development. The domain expert analyses the list of terms elicited in the previous phase for following rules [31]:

i. *Consistency rule*: Any inconsistencies, if present are analyzed. For example usage of different names for the same concept or relationship is an inconsistency which must be identified and removed by the expert.

ii. *Completeness rule*: The incompleteness of the ontology refers to omission of domain concepts and the omission of relationship. This rule checks if the ontology covers all aspects of the domain or not.

If any conflict or disagreement with the rules is detected, the knowledge is refined by the expert.

4.1.2.3. Building Conceptual Model of the Knowledge

This step deals with representing the compiled knowledge in the form of tree, formulas or other structures. This structure is known as the conceptual model of knowledge. The vocabulary identified in the Domain Understanding & Knowledge Elicitation is classified as classes, sub-classes, properties, constraints and individuals. The mechanism can be found in detail in [41]. The class hierarchy is generated in this step using any of the three models top-down, bottom- up or combination of the two [43]. Constraints are also identified on all object and data type properties in this step.

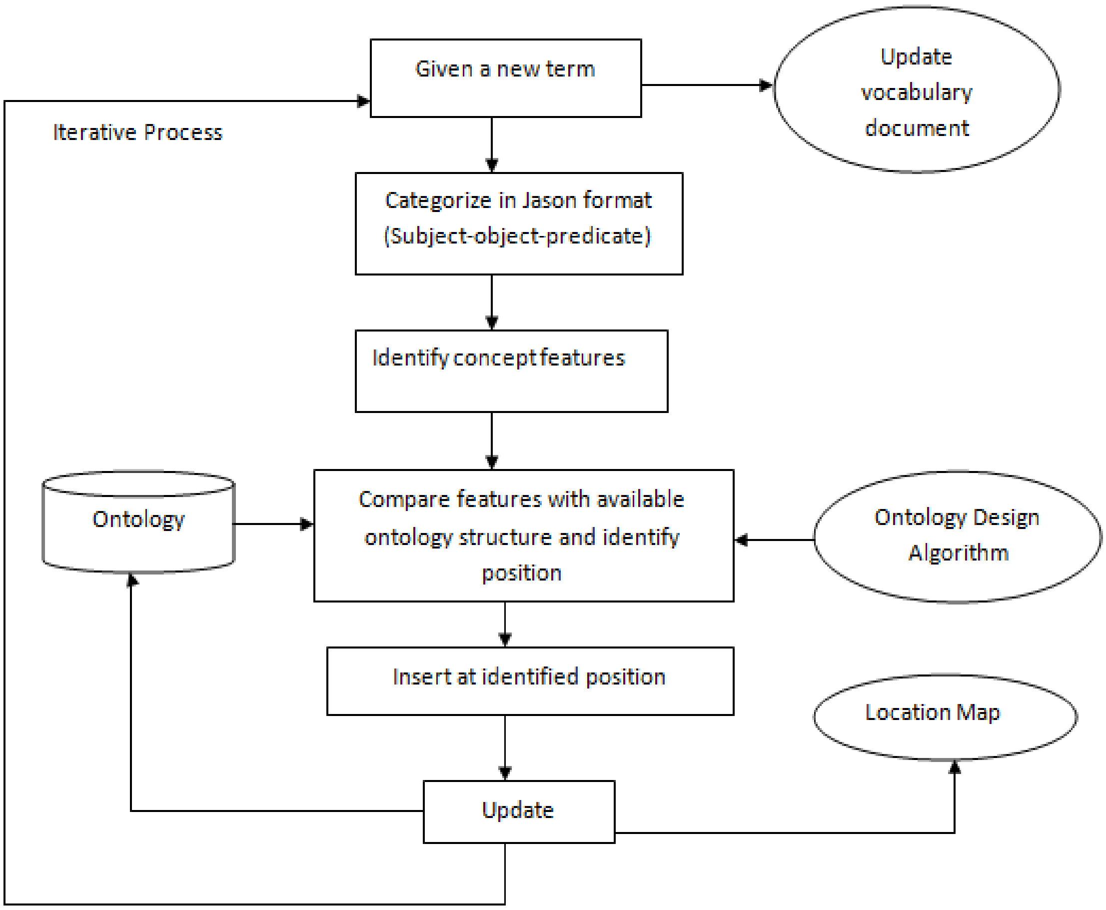
4.1.3. Design Ontology

Analogous to the software engineering design activity, the ontology design phase involves generating the structure of the ontology for supporting its future implementation. In this phase all the knowledge collected from the above steps is put together into a hierarchical structure using a design framework and a location map. Also, an Ontology Design Document (ODD) is produced at the completion of this phase for record keeping, easier understanding and better maintenance

4.1.3.1. Design Framework

We design the ontology using an evolutionary approach. We begin with THING as the root concept and keep on adding concepts one by one into it. Along with this insertion the location map also keeps updating the positions of new concepts. Jason string format is used for the insertion process. The complete design framework is shown in Figure 11 below.

Fig 11. Ontology design framework



4.1.3.2. Design Algorithm

We have automated the above task by implementing the following algorithm:

Design_Ontology(Thing, Concept C1)

i. if (C1 exists in conceptual model)
   continue
   else
   Add concept C1 into conceptual model
ii. Categorize concept C1 in accordance with Jason format: Subject-Predicate-Object.
iii. Insert into table <location map> values <C1>
iv. Identify features of C1: f*(C1)

v. Insert into table <location map> values <f1,f2,f3…….,f_n>

vi. Initialize n=1

vii. For all concepts ‘C’ in location map

Start Loop

For all features of each concept C

Start loop

if (f_n(C1) = = f_n (C))

(a) Mark features f_n (C1) and f_n (C)

(b) n++

else break

End loop

(a) if C1 and C have following feature pattern:

C: f1, f2, f3……….., f_n

C1: f1, f2, f3,………., f_m

Where m>n

if address (C) = (x, y)

then address (C1) = (x+1, *)

(b) else if C1 and C have following feature pattern:

C: f1, f2, f3, f4………..,f_n ; for all n

C1: f1, f2, f3, f4……….,f_m ; for all m

That is, ∍ f_n in C1 for which f_n(C1) != f_n (C)

if address (C) = (x, y)

then address (C1) = (x, *)

(c) else mark C and C1 as “disjoint concepts”.

End Loop

viii. Insert into table <location map> values < address (C1) >

4.1.3.3. Ontology Design Document

Ontology Design Document (ODD) is the output of the Ontology design phase. It helps in keeping record of the formal structure of the ontology and consists of following:

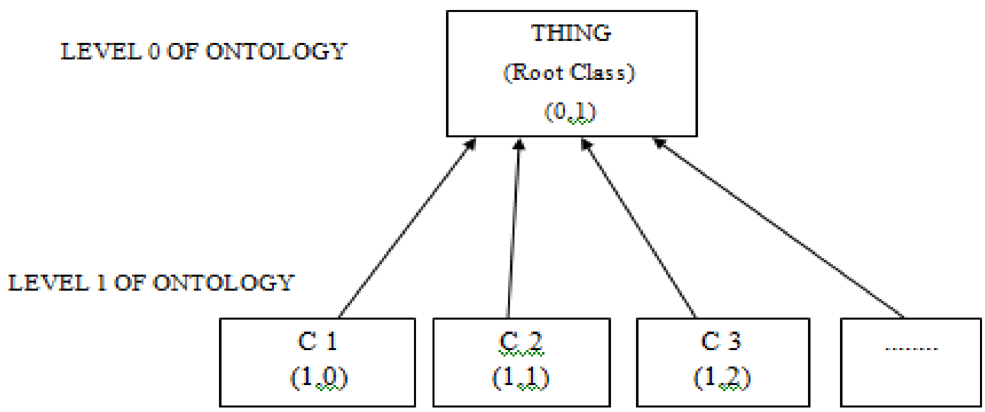
i. *A location map:* Location map is used to build and record the hierarchy of the ontology in a structured and organised manner. It helps in tracking the super class-subclass relationships. Example of a Location map is given in table 2.

CONCEPT NAME	**FEATURE PATTERN** **F(C): <f1,f2,f3,....f_n>**	**CONCEPT ADDRESS** **address(C)**
THING	-	(0,1)
C 1	f 1 , f 2, …...fn	(x,y)
C 2	f 1 , f 2 , …..., fm	(a,b)
…	…	

Table 2. Format of Location Map

ii. *A graphical representation of the ontology structure:* A layout of the graphical representation used in the ODD is shown in Figure 12.

Fig 12. Graphical representation of Ontology

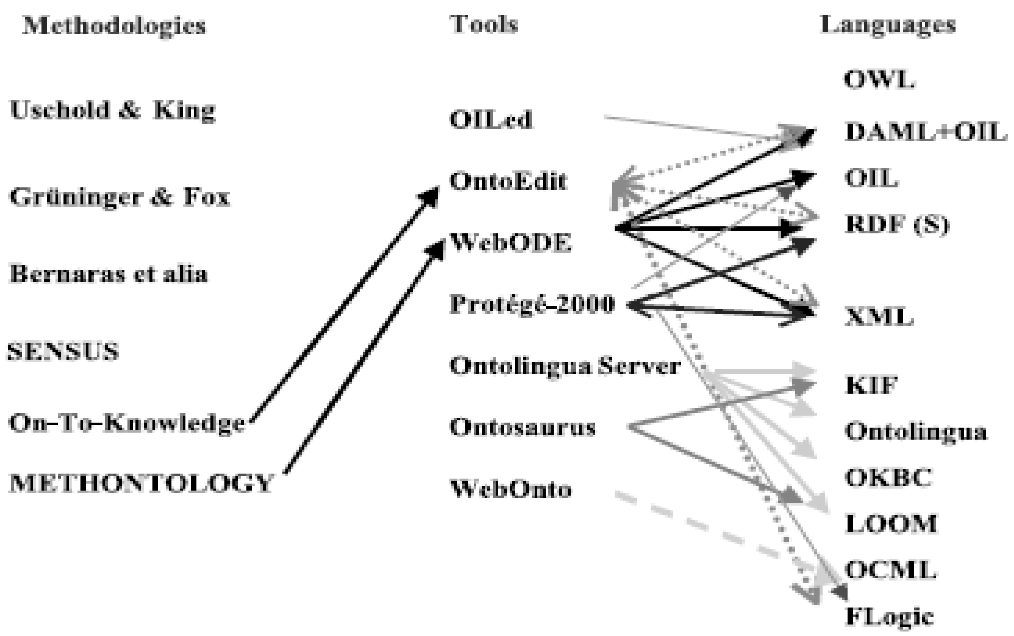


4.1.4. Formalization/ Implementation

This phase deals with implementation of the above mentioned ontology design using any formal ontology development language and tools.

Many ontology languages (RDF/RDFS, OWL etc.), editing tools (Protégé, OntoEdit etc.) and standards (OWL-S etc.) are available for supporting this task. The most popular and recently developed standard ontology language is OWL [47]. OWL is the most academically and commercially popular ontology language used for knowledge representation. It is endorsed by World Wide Web Consortium (W3C). Details of ontology development languages have been given in chapter 1. Any of the available languages can be used to implement the ontology. If it is not necessary to implement the ontology manually any of the above mentioned tools can be used. Figure 13 shows the languages and tools used for ontology formalization in different researches, as discussed in Chapter 2.

Fig 13. Tools and Languages used for Ontology Formularization



We use the tool Protégé as the ontology editor because it is commercially most popular, easily available and user friendly. In this work the ontology will be implemented in OWL language because it is easier to understand and supported by protégé. More details of this step of development are discussed in the next chapter with the help of a case study on the Hospital (medical healthcare) domain.

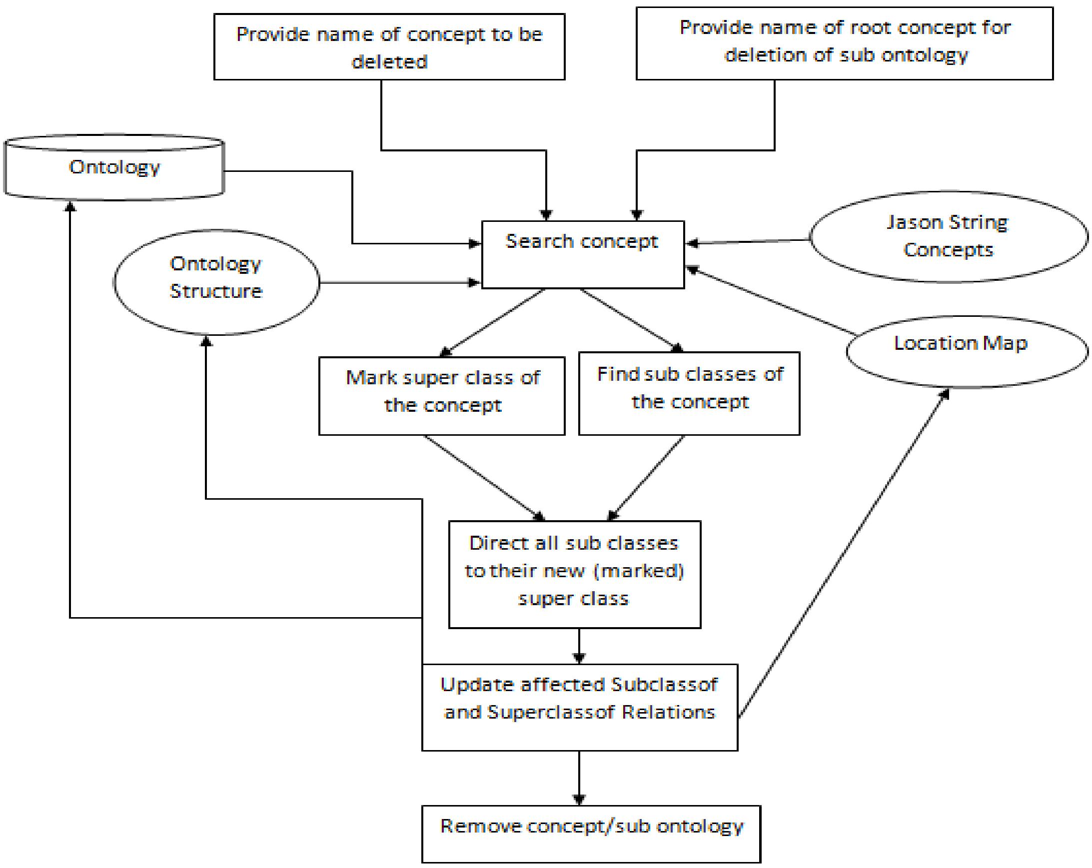
4.1.5. Maintenance

Due to changing requirements and changing specifications the ontology must be modified and maintained. These modifications and adaptations are performed under the maintenance phase. This step deals with adding, updating and removing concepts/ parts/ individuals of the ontology. This phase is broadly classified into following categories:

i. *Addition of new concept*: Technique as used in ontology design can be used, as shown in Figure 8.

ii. *Deletion of a concept:* Technique as shown in Figure 14 can be used [44].

Fig 14. Deletion of a concept or sub ontology



4.1.6. Evolutionary Activities

These activities are performed simultaneously with every phase of ontology development. These activities are following:

4.1.6.1. Evolving Knowledge Acquisition

This step involves acquiring knowledge throughout the development process. This newly acquired knowledge is also included in the existing model of ontology for the purpose of *better domain cognition*.

4.1.6.2. Integration

It involves merging concepts from other ontologies or merging other ontologies as a whole with the current ontology as and when needed.

4.1.6.3. Documentation

Extensive documentation is performed along with each phase of the enhanced On-To-Methodology which supports better traceability, maintenance and error correction. The examples of such documents are Ontology Specification Document (OSD), Ontology Design Document (ODD), Ontology Evaluation Report etc. Among these, ODD has been discussed in the previous sections and the others will be discussed in further chapters.

4.1.6.4. Evaluation

After the construction of the ontology, it is evaluated for its efficiency on different basis. The term ‘Evaluation’ encompasses both- verification & validation. In this approach a threefold evaluation is recommended which includes following:

i. *Manual Evaluation*: This kind of evaluation totally depends on the experts’ knowledge. In this substep the expert uses his personal experience and knowledge to evaluate the ontology.

ii. *Semi Automatic Evaluation*: This evaluation is partly automated using Protégé plug ins and partly uses the knowledge of the developer/expert. This step can be sub divided into 2 sub steps [32] namely, *Relation Evaluation* and *Hierarchy Evaluation.* Relation Evaluation is done using the *Reasoner* that comes inbuilt in the tool, like Fact++ in Protégé. Hierarchy Evaluation makes use of ontology tool (like OntoGraf, OWLViz) to infer and create Ontology Graph automatically, and judge whether Subordination between the class and individual in the graph are coincident.

iii. *Automatic Quality Evaluation*: The evaluation of quality of the ontology is performed by generating the values of Ontometric metrics for the ontology. This step is fully supported by the automation developed for the approach. But this evaluation is not performed parallely. *It is only performed after the implementation process is completed.*

The output of this phase is the *Evaluation Report*. It consists of following:

i. Evaluation results of competency questions asked by the experts from *manual evaluation.*

ii. Evaluation results of OWLViz and OntoGraph from *semi automatic evaluation.*

iii. Evaluation results provided by the tool, including the graphical results and values obtained for Ontometric characteristics. This is discussed in detail in Chapter 5.

4.1.6.5. Configuration Management and Integration

This phase ensures that any changes made in the ontology will not affect its consistency. After the execution of the evaluation phase, the conflicts are resolved by making changes into the conceptual model or the developed ontology. These changes can affect other aspects of the ontology, such as adding or deleting or relocating of a concept may require adaptations by corresponding relationships also.

Chapter 5: IMPLEMENTATION

This chapter discusses the implementation of the proposed methodology in detail. The implementation work done for this project can be well described in two sections. The first section presents the implementation details of the hospital ontology and the second section presents implementation details of the developed tool.

5.1. Implementation of the Hospital Ontology

A Hospital Ontology was developed as a case study for this project to formalize the methodology proposed. The open source tool Protégé was used for the construction of the Hospital Ontology using On-To-Methodology as the development methodology. Protégé is a standard tool [27] provided by the Stanford University and is used by most of the Ontology Engineering community. It is used for creation of ontology in languages such as OWL, RDF etc. This tool provides many other functions that can be applied on the ontologies such as the extension of the ontology etc. Many plug-ins are available for this tool. Protégé provides a built in reasoner such as Fact++ [47] or Pellet [48] which can be used to generate the inferred hierarchy of the ontology domain.

5.1.1. Steps involved in construction of Hospital Ontology

Following steps were followed for construction of Hospital Ontology:

Step 1: Preliminary Specifications

- *Objective*: In this step the objective of ontology development is identified. The objective of Hospital Ontology can be stated as follows:

"The aim of the Hospital Ontology is to answer the queries of customers across a large information base of a hospital or group of hospitals, based on multiple search criteria with complex inter-relations."

- *Domain*: Medical Healthcare Domain (specifically single or group of hospitals)
- *Viewpoints/ Stakeholders*: Customers who want any information about the hospital, hospital staff, Patients, hospital management etc.

Step 2: Knowledge Acquisition

- *Domain Understanding & Knowledge Elicitation:* The vocabulary of the hospital domain is identified in this phase. Due to the huge collection of terms only the main terms identified in this phase are listed below.

  Medicines, Infrastructure, Department, Building, Medical Conditions, Allergies, Disease, Medical History, Symptoms, Antibiotics, Analgesics, Antipyretics, Antiseptics, Test, Treatment, Medical _Personnel, Employee, Person , Patient, Doctor, Nurse, etc.

- *Analysis of knowledge and conflict resolution:* The completeness and consistency of the identified vocabulary is checked in this step and the redundant terms are removed. For example, *leukemia* is also known as *blood cancer* or *bone marrow cancer. So blood cancer and bone marrow cancer terms are removed from the vocabulary.*

- *Building Conceptual Model of the Knowledge:* The compiled terms that result after the above phase are categorized into classes, subclasses, properties, constraints and individuals in this phase. We use *top-down* strategy for this conceptualization. Table 3 presents the Superclass-Subclass relationships identified for the conceptual model.

Super-class	**Subclasses**
Thing (Root Class)	Department, Field, Infrastructure, Medicines, Medical conditions, Medical History, Person, Symptoms, Tests, Treatment
Infrastructure	Buildings

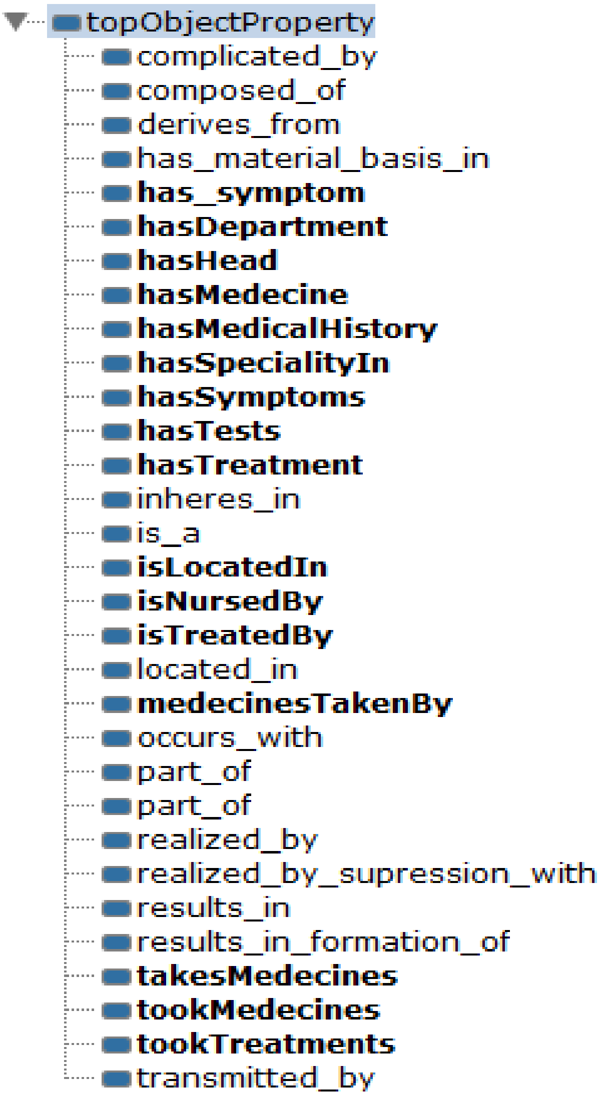
Medicines	Analgesics, Antibiotics, Antimalarial_drugs, Antipyretics, Antiseptics, Mood_stablizers, Oral_contraceptives
Medical Conditions	Allergies, disease
Disease	Disease by infectious agents, disease of anatomical entity, disease of cellular proliferation, disease of mental health, disease of metabolism, genetic disease, physical disorder, syndrome etc.
Person	Employee, Patient
Employee	Doctor, Nurse, Other_hospital_personnel
Patient	In_Patient., Out_Patient
Symptom	Abdominal symptom, cardiovascular system symptom, digestive system symptom, general symptom, head and neck symptom, nervous system symptom, urinary system symptom, reproductive system symptom, skin and integumentary tissue symptom, hemic and immune system symptom, respirator and chest system symptom, nutrition symptom, metabolism and development symptom etc.
Tests	Vidal test, acidification test, hormone test, White cell count, Red cell count, Full cell count, skin biopsy, HIV antigen test, Thyroid test, glucose test, Insulin test, Lipids, MCV, Malaria antigen test, Nerve biopsy, parasite test etc.
Treatments	Chemotherapy, antibiotic therapy, antimicrobial therapy, antiviral therapy,

	Surgery, Conservative rehabitation, Pain management treatment etc.

Table 3. Top level class hierarchy of the Hospital Ontology

The object properties identified are shown in Figure 15.

Fig. 15 The property hierarchy of the Hospital Ontology



The root of the property hierarchy is named as Top Object property and all the child properties are listed below it. Figure 16 shows an example of hierarchy of data properties in the hospital ontology.

Fig 16. Example of hierarchy of data properties

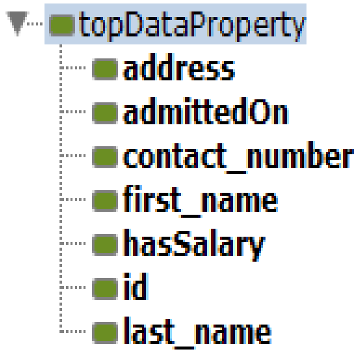


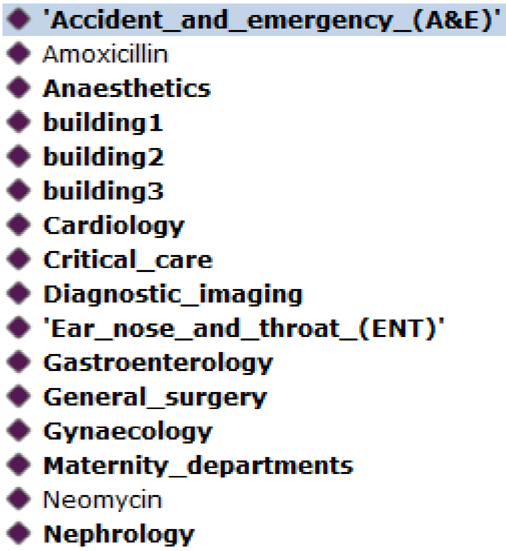
Table 4 shows the domain and ranges of some of the properties as defined above. Domain and Range are used for setting up constraints on the applicability of the properties.

Property	Domain	Range	Type of Property
has_symptom	Disease	Symptom	Object property
hasDepartment	Nurse Doctor Other_hospital_personnel	Department	Object property
hasSpecialityIn	Nurse Doctor	Field	Object property
hasHead	Department	Employee	Object property
isTreatedBy	Patient	Doctor	Object property
tookMedecines	Medical_History	Medecines	Object property
address	Person	String	Data property
admittedOn	Patient	dateTime	Data property
hasSalary	Employee	Integer	Data property

Table 4. Constraints on properties in Hospital Ontology

Along with the above, inverse relations among the properties are also identified. For example, *takesMedecines* property is Inverse of *medecinesTakenBy* property. Figure 17 shows some of the individuals identified in the Hospital ontology.

Fig. 17 Individuals in Hospital Ontology



- **ONTOLOGY SPECIFICATIONS DOCUMENT (OSD) for hospital ontology**

Domain: Hospitals/ Healthcare/ Medical

Author/Authors: Anupriya Tewari

Date: 15 May 2014

Place: New Delhi, India

1. Introduction

This document aims at defining the complete requirements for a 'Hospital Ontology'. The final Ontology must have the features/functionalities mentioned in this document and no assumptions should be made for any supplementary functionality by any of the parties involved in development/testing/implementation/usage of this ontology. In case any addition features are required from the product, a formal request for updation must be made and subsequently a new release of this document and/or product will be produced.

1.1. Purpose

The purpose of this document is to record the requirements for the hospital ontology and provide a basis for verification at later stages of development. This document also serves as the starting point of the ontology design phase.

1.2. Scope

The intended Hospital Ontology will act as a semantic knowledge base for inferring any kind of information from the domain of hospitals, thereby providing efficient services for customers/patients of the hospital as well as the hospital staff/employees. This ontology will provide following services to its users:

- This ontology will hold information to answer queries of customers based on single or multiple parameters, such as, disease, symptoms, allergies, availability of doctors/medicines etc.
- Hospital authorities can use this information to check the status and growth of different departments in the hospital and thus encourage research and experiments in the weak departments.

1.3. Definitions, acronyms and abbreviations

Employee: Employee is a person who works for the hospital and receives salary for it, for example, nurses, doctors, compounders etc.
Customer: A person who desires any kind of information from the ontology about the hospital.

Patient: A person who is currently admitted in the hospital or who has applied for admission in the hospital or who has a medical history with the hospital.

OWL: Web Ontology Language

HIV: Human Immunodeficiency Virus

AZT: Azidothymidine

And so on..

1.4. References

Document 1: Elicited vocabulary as discussed in previous section 4.1.1, step 1.

Document 2: Extracted concepts from domain understanding and knowledge elicitation

Document 3: Domain vocabulary after redundancy removal

Document 4: Conceptual model of knowledge

1.5. Sources of knowledge

- Books: Principles of hospital administration and planning, information technology for the healthcare professionals.
- Experts: Ms. Anupriya Tewari, Ms. Neha Nagpal, Ms. Dhwani Dholakia
- Websites: http://www.rcpamanual.edu.au

  http://emedicine.medscape.com

  http://www.nice.org.uk etc.

- Other Ontologies: DOID.owl, SYMP.owl etc.

1.6. Overview

Section 2 of this document describes the overall ontology in terms of its characteristics, intended users, functions, constraints etc. Section 3 describes in detail functional as well as non-functional requirements of the system.

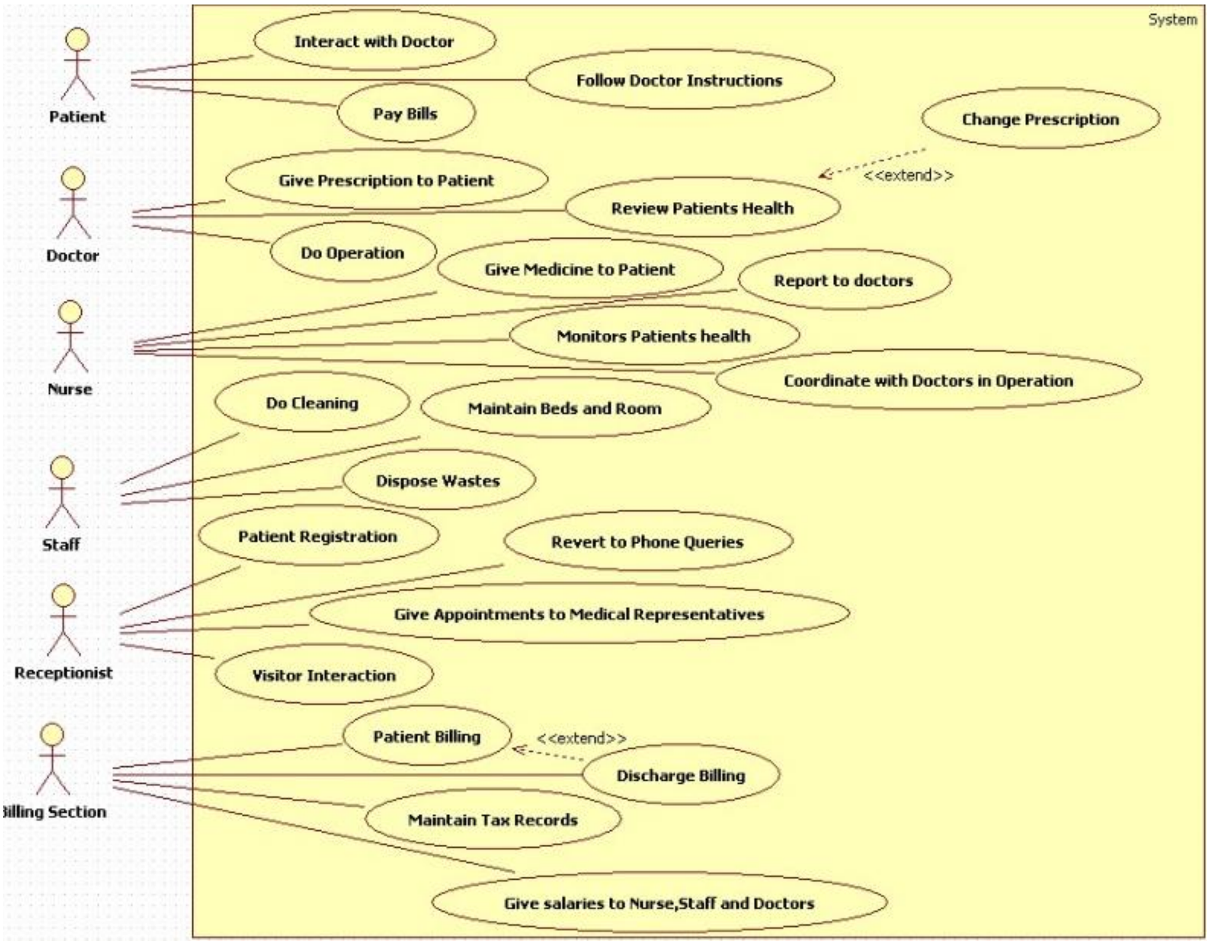
2. Overall Description

The Hospital ontology will be a representation of a hospital in the domain of ontology. Any users of this product can query the ontology to receive information regarding the availability of doctors/departments/rooms, available treatments or tests etc.

2.1. Ontology Perspective

This system will be an independent software product having the ontology as the backend and along with a front end which will be discussed in the next section. The use case diagram for a hospital is shown in figure 18.

Fig. 18. Use Case Diagram for a Hospital



2.2. Ontology Functions

- Answer all queries of the customers.
- Help the hospital management in better management of the hospital.
- Help the hospital staff in tracking and recording information about all employees.

And so on.

2.3. User Characteristics

Users of this ontology are the hospital staff, hospital management and other general users. Assuming, they don't have much knowledge of ontologies the system must have a well designed UI.

- Educational Qualification: Bachelor in engineering/technology/science.

- Technical Expertise: Knowledge of OWL and Protégé.

2.4. Constraints

The users can explore information of only one hospital by this product. Moreover no security measures have been incorporated in the ontology, so any user can make changes in it.

3. Specific Requirements

This section defines the requirements in more detail.

3.1. Interfaces

i. Hardware Interface: Screen resolution of at least 1024 X 768.
ii. Software Interface: Any operating system (Windows XP, Vista, 7, Mac OS, Linux)
iii. Communication Interfaces: None
iv. Memory constraints: At least 512MB RAM for running the web browsers.

3.2. Functional Requirements: As discussed in section 2.2.
3.3. Performance Requirements: Response time for customer queries must be less than 10 seconds.
3.4. Other requirements: None

Step 3: Design Ontology

The Hospital Ontology was designed using the algorithm previously described in section 4.1.3.2. The ontology design document (ODD) for the Hospital Ontology is discussed in this section.

- **ONTOLOGY DESIGN DOCUMENT FOR HOSPITAL ONTOLOGY**

This ODD represents the design of the Hospital Ontology in two parts. Firstly, the location map and secondly, the graphical representation. A section of the location map is presented in Table 5. Due to huge depth of the Hospital Ontology only the high level concepts are shown in Table 5.

Concept Name	Feature Pattern	Concept Address
Thing	-	(0,1)
Department	f^+1	(1,1)
Field	f^+2	(1,2)
Infrastructure	f^+3	(1,3)
Medicines	f^+4	(1,4)
Medical conditions	f^+5	(1,5)
Medical History	f^+6	(1,6)
Person	f^+7	(1,7)
Symptom	f^+8	(1,8)
Tests	f^+9	(1,9)
Treatment	f^+10	(1,10)
Buildings	$f^+3 \bullet f^+11$	(2,1)
Analgesics	$f^+4 \bullet f^+12$	(2,2)
Antibiotics	$f^+4 \bullet f^+13$	(2,3)
Antimalarial_drugs	$f^+4 \bullet f^+14$	(2,4)
Antipyretics	$f^+4 \bullet f^+15$	(2,5)
Antiseptics	$f^+4 \bullet f^+16$	(2,6)
Mood_stablizers	$f^+4 \bullet f^+17$	(2,7)
Oral_contraceptives	$f^+4 \bullet f^+18$	(2,8)
Allergies	$f^+5 \bullet f^+6 \bullet f^+8 \bullet f^+19$	(2,9)
Disease	$f^+5 \bullet f^+6 \bullet f^+8 \bullet f^+20$	(2,10)
Disease by infectious agent	$f^+5 \bullet f^+6 \bullet f^+8 \bullet f^+20 \bullet f^+21$	(3,1)

Disease of anatomical entity	$f^+5 \bullet f^+6 \bullet f^+8 \bullet f^+20 \bullet f^+22$	(3,2)
Disease of cellular proliferation	$f^+5 \bullet f^+6 \bullet f^+8 \bullet f^+20 \bullet f^+23$	(3,3)
Disease of mental health	$f^+5 \bullet f^+6 \bullet f^+8 \bullet f^+20 \bullet f^+24$	(3,4)
Disease of metabolism	$f^+5 \bullet f^+6 \bullet f^+8 \bullet f^+20 \bullet f^+25$	(3,5)
Genetic disease	$f^+5 \bullet f^+6 \bullet f^+8 \bullet f^+20 \bullet f^+26$	(3,6)
Physical disorder	$f^+5 \bullet f^+6 \bullet f^+8 \bullet f^+20 \bullet f^+27$	(3,7)
Syndrome	$f^+5 \bullet f^+6 \bullet f^+8 \bullet f^+20 \bullet f^+28$	(3,8)
Bacterial infectious disease	$f^+5 \bullet f^+6 \bullet f^+8 \bullet f^+20 \bullet f^+21 \bullet f^+29$	(4,1)
Commensal Bacterial infectious disease	$f^+5 \bullet f^+6 \bullet f^+8 \bullet f^+20 \bullet f^+21 \bullet f^+29 \bullet f^+30$	(5,1)
Pertusis	$f^+5 \bullet f^+6 \bullet f^+8 \bullet f^+20 \bullet f^+21 \bullet f^+29 \bullet f^+30 \bullet f^+31$	(6,1)
Bordetella parapertussis whooping cough	$f^+5 \bullet f^+6 \bullet f^+8 \bullet f^+20 \bullet f^+21 \bullet f^+29 \bullet f^+30 \bullet f^+31 \bullet f^+32$	(7,1)

Table 5. High level concepts in Location map of Hospital Ontology

In table 5, f^+n represents a feature number. The features which are inherited by a class from its super-class are highlighted. All the first level concepts are annotated with single features. More clearly, f^+1 represents all features combined together that represent a department, similarly, f^+4 represents all the features combined together that can define medicines. Thus, antibiotics are a sub-class of medicine class, because they have $f^+4 \bullet f^+13$ as their feature. Same concept is followed for all other classes.

Figure 19 shows the graphical representation of the high level concepts at the design phase. The Concepts highlighted in gray represent that they are further organised into sub-classes. Some of the sub-ontologies rooted at these classes are shown in figure 20and 21 and 22.

Fig 19. Graphical structure of high level concepts in Hospital Ontology

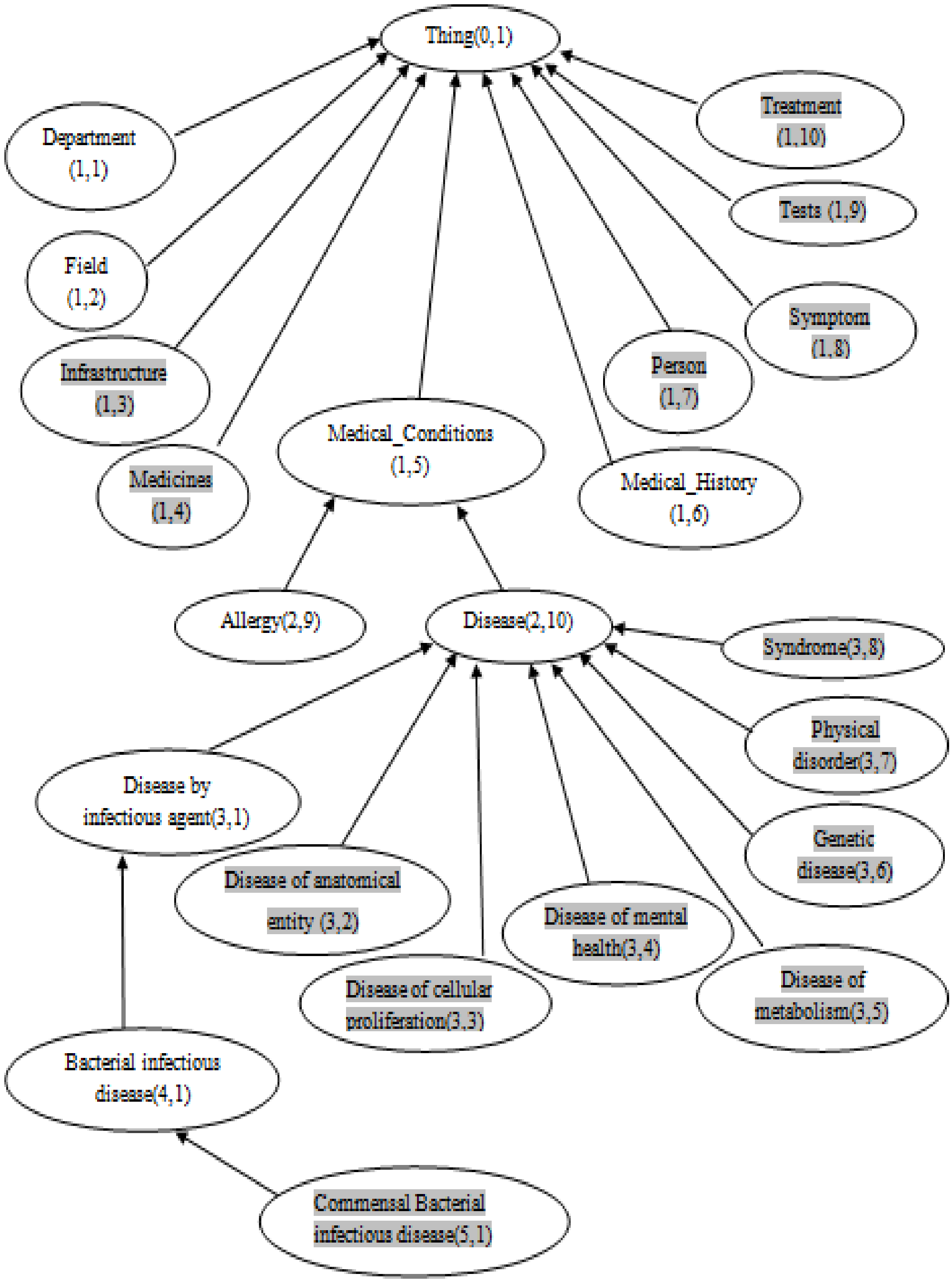


Fig 20. Sub-Ontology rooted at Infrastructure

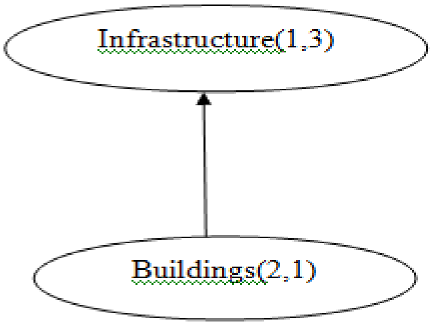


Fig 21. Sub-Ontology rooted at Medicine

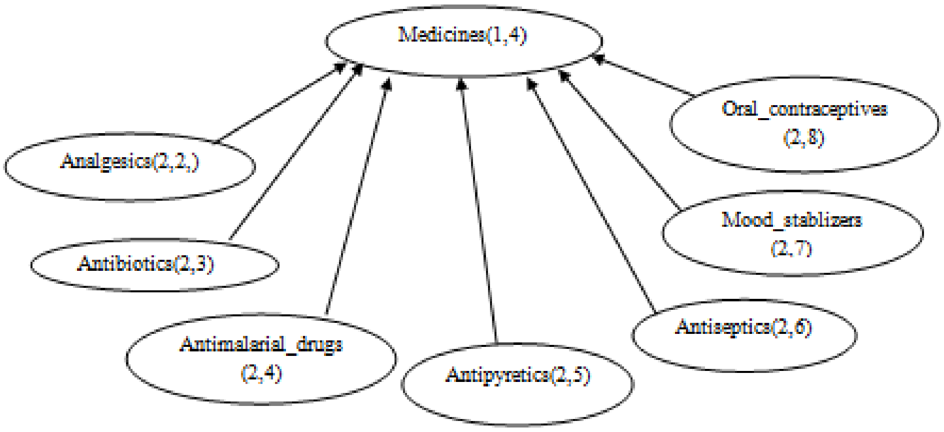
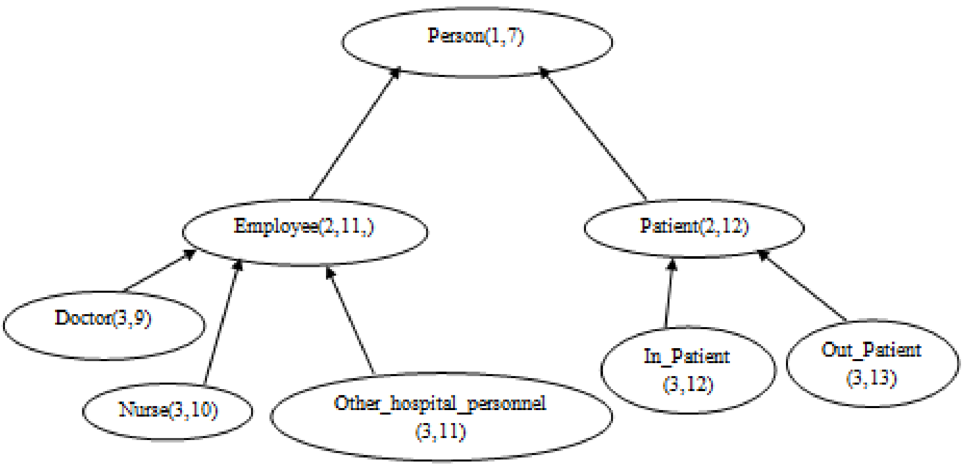


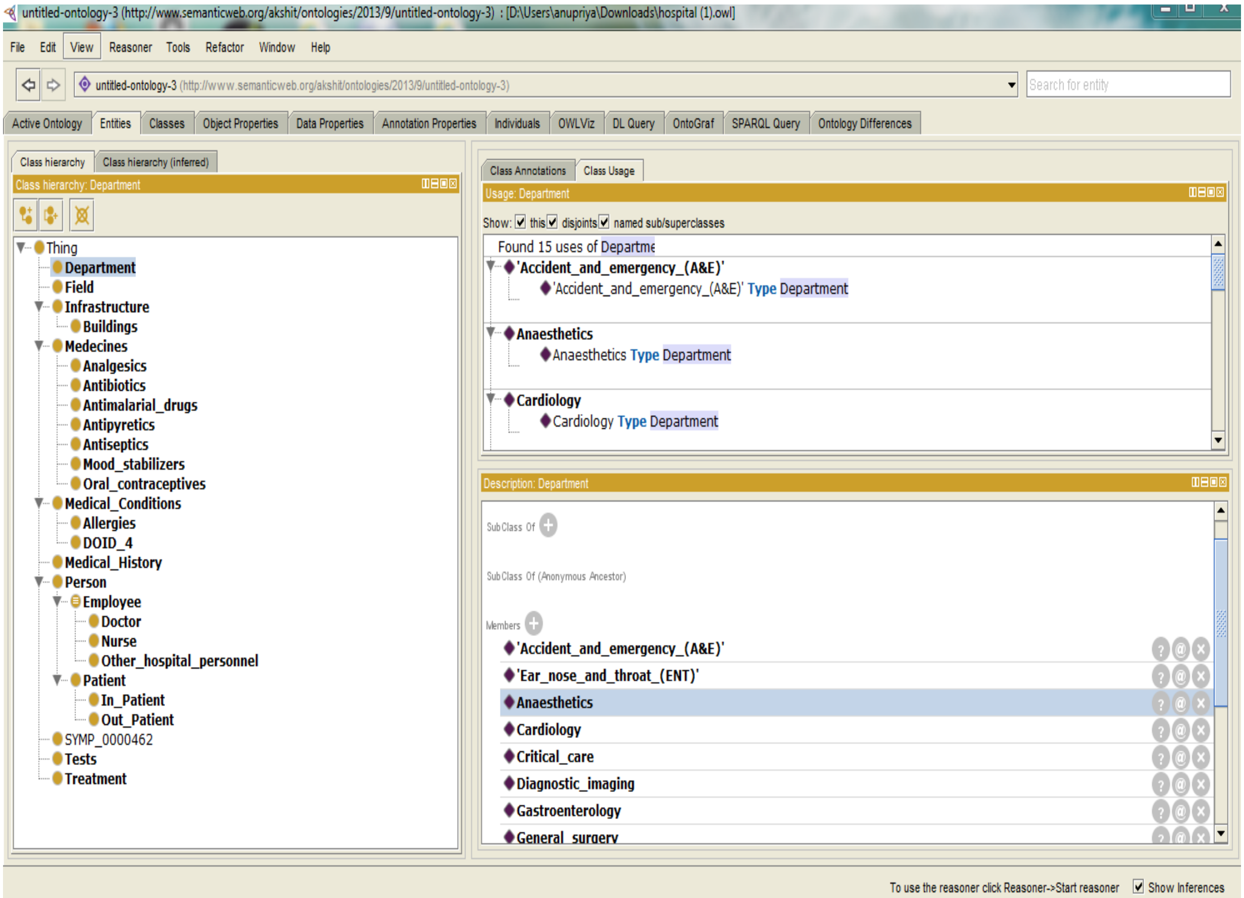
Fig 22. Sub-Ontology rooted at Person



Step 4: Formalization/Implementation

The hospital ontology was formalized using Protégé tool and OWL as the ontology implementation language. Figure 23 and 24 show the snapshots of the Hospital Ontology on Protégé.

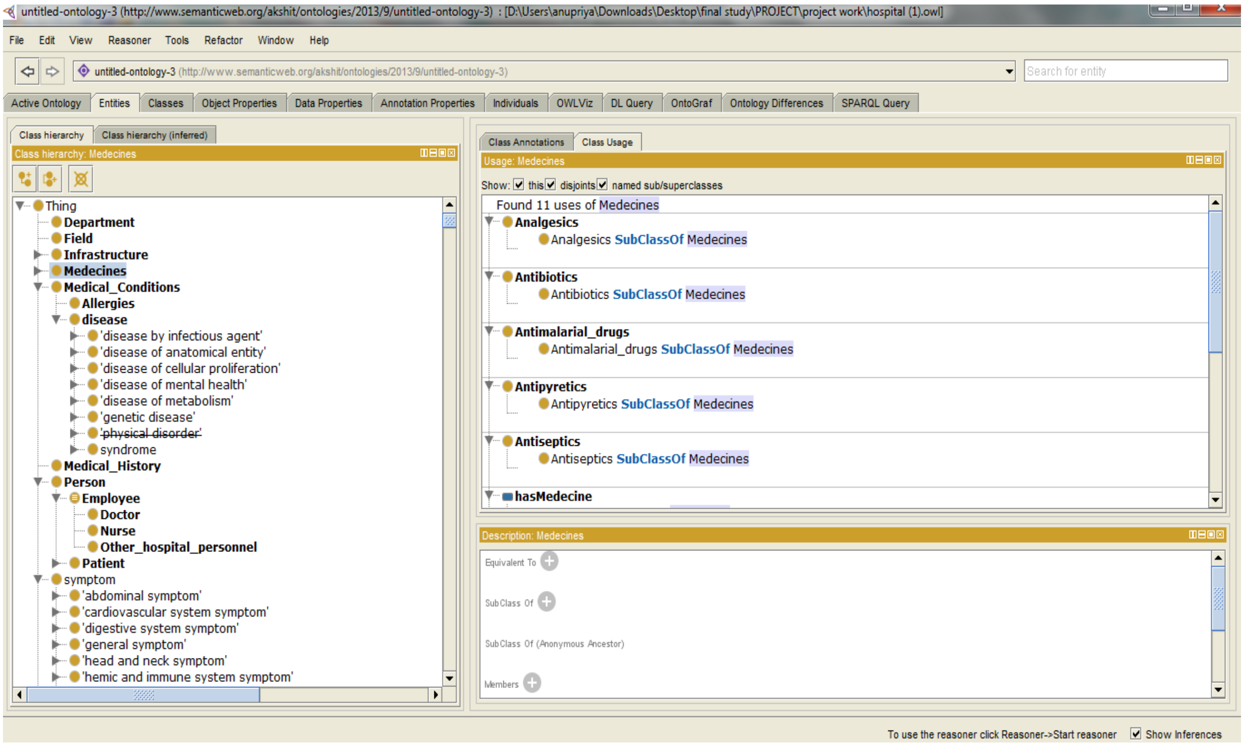
Fig 23. Class hierarchy of Hospital Ontology on the tool Protégé (offline version)



In the Figure 23 the imported ontology files of the DOID ontology and the SYMP ontology are not loaded, hence this is shown as the offline version of the ontology.

Figure 24, shows the class hierarchy of the Hospital Ontology after the doid.owl and symp.owl files have been imported online, thus it is named as the online version of the hospital ontology.

Fig 24. Class hierarchy of Hospital Ontology on the tool Protégé (online version)



Step 5: Evaluation and Maintenance

Evaluation of the Hospital ontology for completeness and consistency checking is described in detail in Chapter 5. Maintenance of the Hospital Ontology mainly consists of addition and deletion of concepts, which is automated by the developed Tool and is discussed in the next section.

5.2. Implementation of the Tool

The enhanced On-To-Methodology tool is developed to provide automated support to the proposed methodology. This tool was employed in the Ontology Design activity to design the ontology automatically. It was developed using PHP as the implementation language and uses SPARQL queries to infer knowledge from the ontology. The following frameworks have been used to implement the tool:

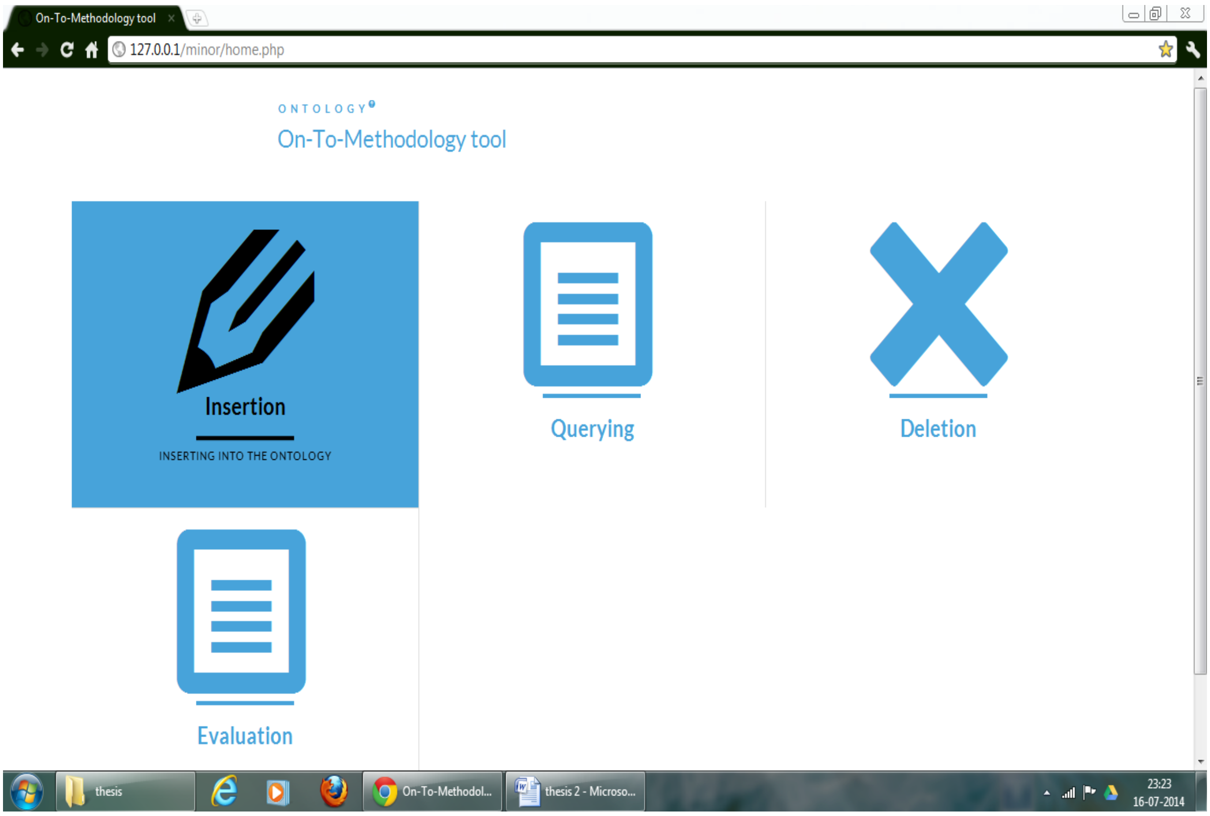
- WAMPP SERVER: for running PHP commands on the server.
- VIRTUOSO Framework: for generating the SPARQL End Point.
- SPARQL QUERIES
- CSS code for designing the tool's front page.
- HTML coding for other functions.

The tool provides following 4 functions:

- *Insertion* of concepts/individuals in the ontology
- *Deletion* of concepts
- *Query* the Subject-Predicate-Object Triplet from the ontology
- *Evaluation* of the linked Hospital ontology by comparing it with another ontology for hospitals, which is constructed using another methodology METHONTOLOGY.

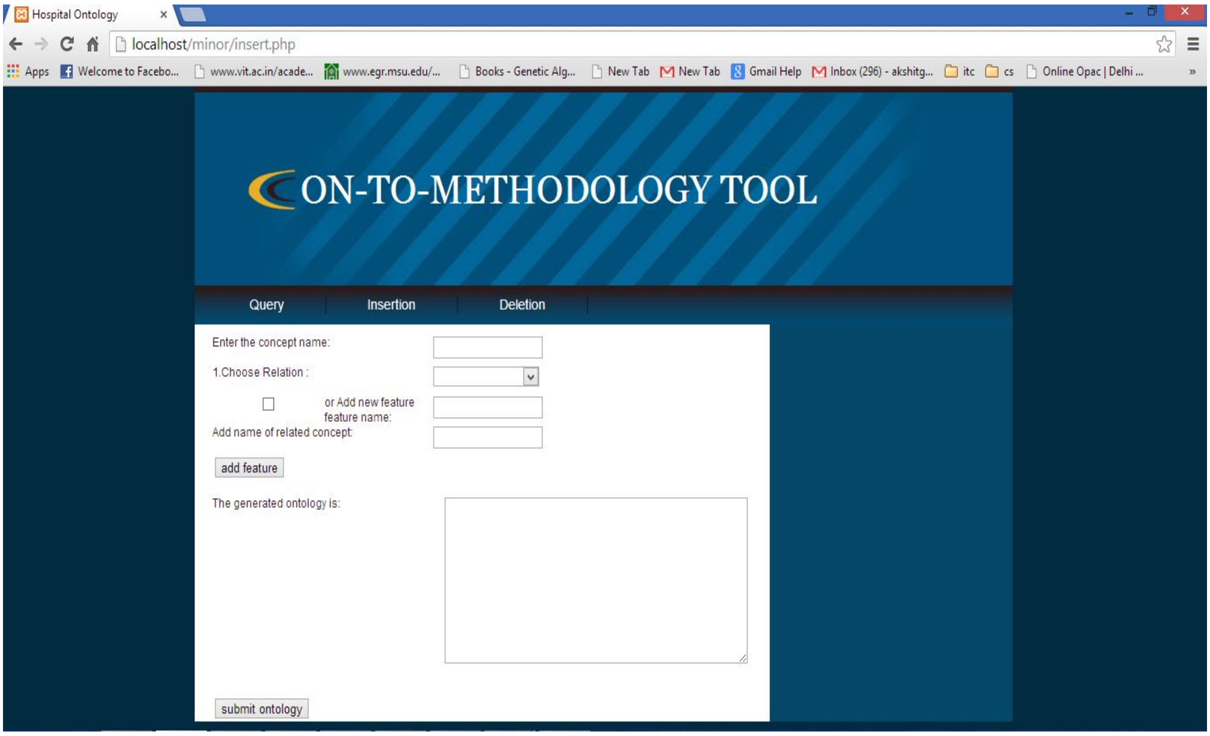
Amongst these 4 functions only first 3 will be discussed in this chapter. Evaluation will be discussed in the next chapter. Figure 25 shows the home page of the Tool.

Fig 25. Home Page of On-To-Methodology Tool



The following figures show the employment of the Tool for design of the Hospital Ontology. Figure 26 shows the snapshot of the tool for its *Insertion* functionality.

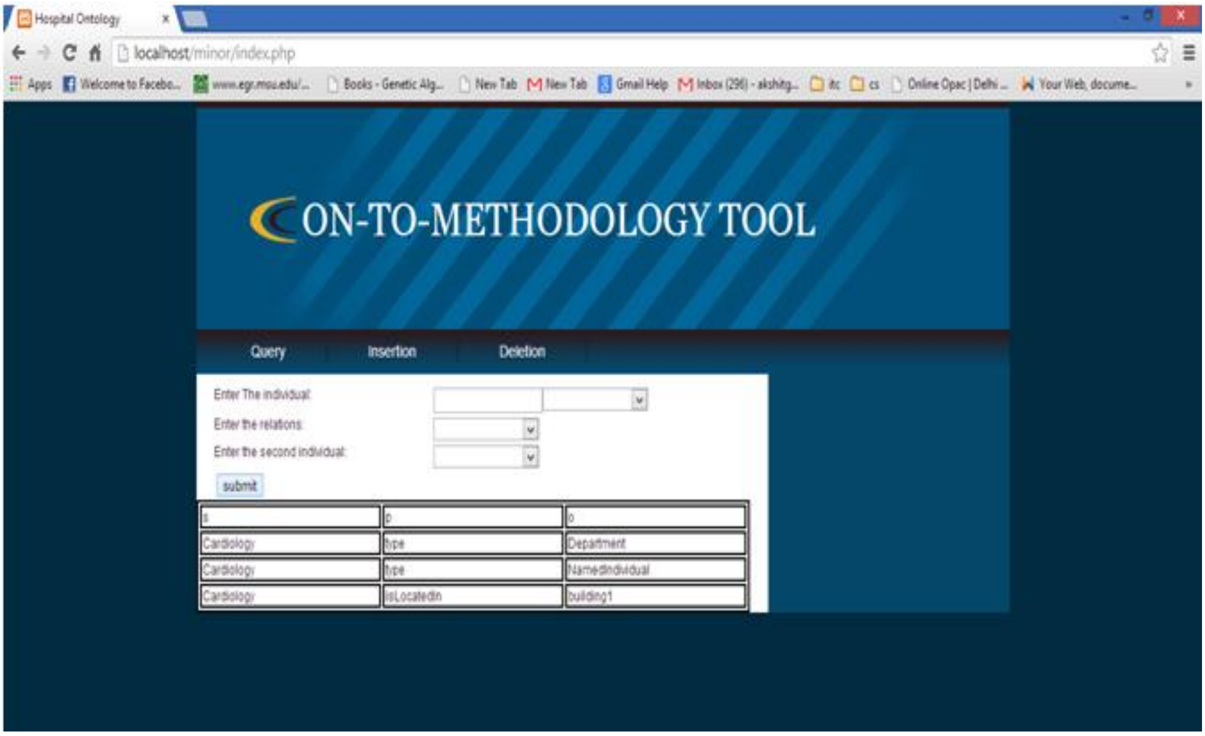
Fig 26. On-To-Methodology tool: Insertion of new Concept



- The concept name to be inserted is entered in the provided text field of the insertion tab of the Tool.
- The next step is to provide a relationship that this concept may have with any other concept existing in the ontology.
- This relation can be newly defined or can be chosen from a drop down menu.
- Then the concept name with which the inserted concept is related is also mentioned.
- Finally the ontology is submitted and can be seen in the given text field.

Figure 27 shows the Query Functionality of the tool.

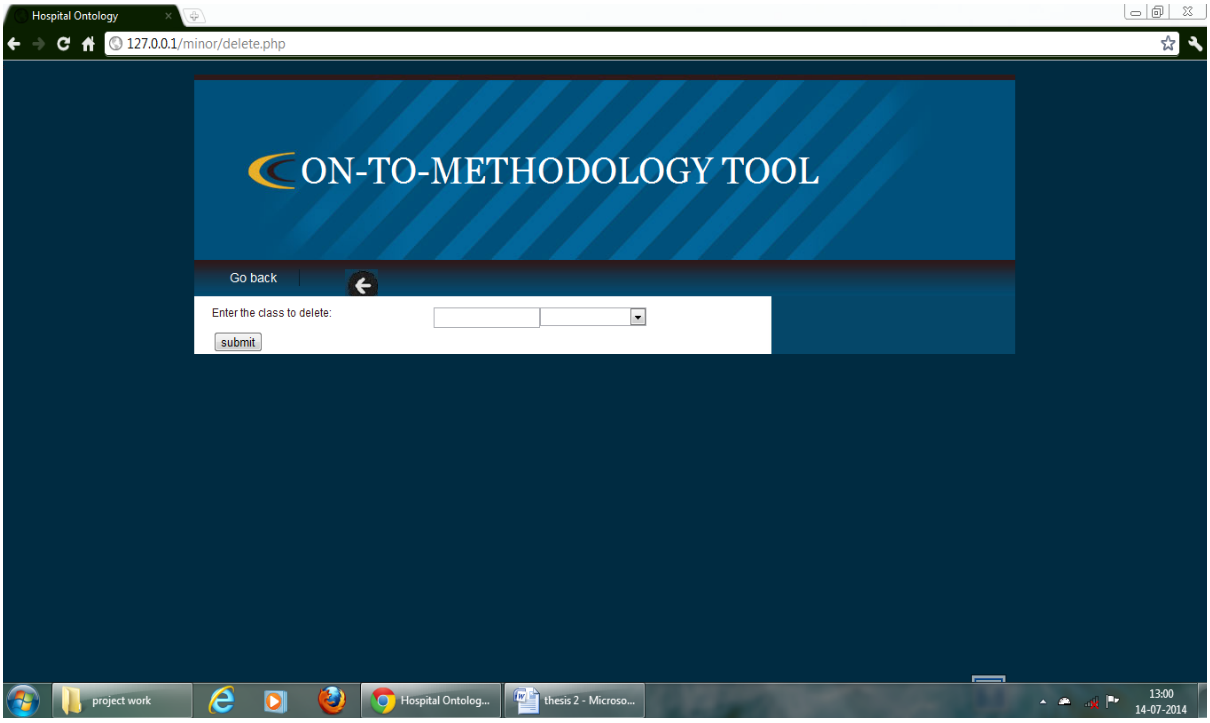
Fig 27. On-To-Methodology Tool: Query



- Firstly, the name of the individual about which the query is made is typed in the first text area.
- Then the name of any relation it might be having with any other individuals will be typed in the next text box. . If no such information is available, “other” is chosen from the drop down menu.
- The name of the individuals with which the queried individual is having relations, is also mentioned. If no such information is available, “other” is chosen from the drop down menu.
- Finally, the submit button fires the query.
- The results of the query are shown in the form of a table, which provide all subject-object-predicate groups that involve the queried individual as the subject.

Figure 28 shows the Deletion Functionality of the tool.

Fig 28. On-To-Methodology Tool: Deletion



The name of the concept to be deleted is typed in the text field and the change is submitted. The related changes are automatically made by the tool.

Chapter 6: EVALUATION AND RESULTS

This chapter presents the Evaluation section of the development methodology. This chapter is divided into four sections namely, semi automatic evaluation of the Hospital Ontology, subjective evaluation of the enhanced On-To-Methodology, automatic evaluation of the enhanced On-To-Methodology and results.

6.1. Semi automatic evaluation of the Hospital Ontology

In this step we evaluate the developed Hospital Ontology using three different methods, including the following:

i. Evaluation by competency questions

Competency questions are a recommended way to define scope and afterwards validate the ontology. Competency questions are questions written in natural language by domain experts that the ontology should be capable to answer.

The evaluation checks the ability of the ontology to express the competency questions and their corresponding answer statements, and the simultaneous search for more information that can help to build up the ontology. The developed ontology was evaluated by experts by querying it with competency questions and verifying the results. The DLQuery tab of Protégé tool was used for this step.

Figure 29 shows the results of evaluation performed using the following query:

"What are the diseases that *hasSymptom chills* and *shakes* and *'profound weakness'* and that *hasMedecine Antibiotics* and *Antipyretics* ."

Fig 29. Evaluation of competency question: DLQuery

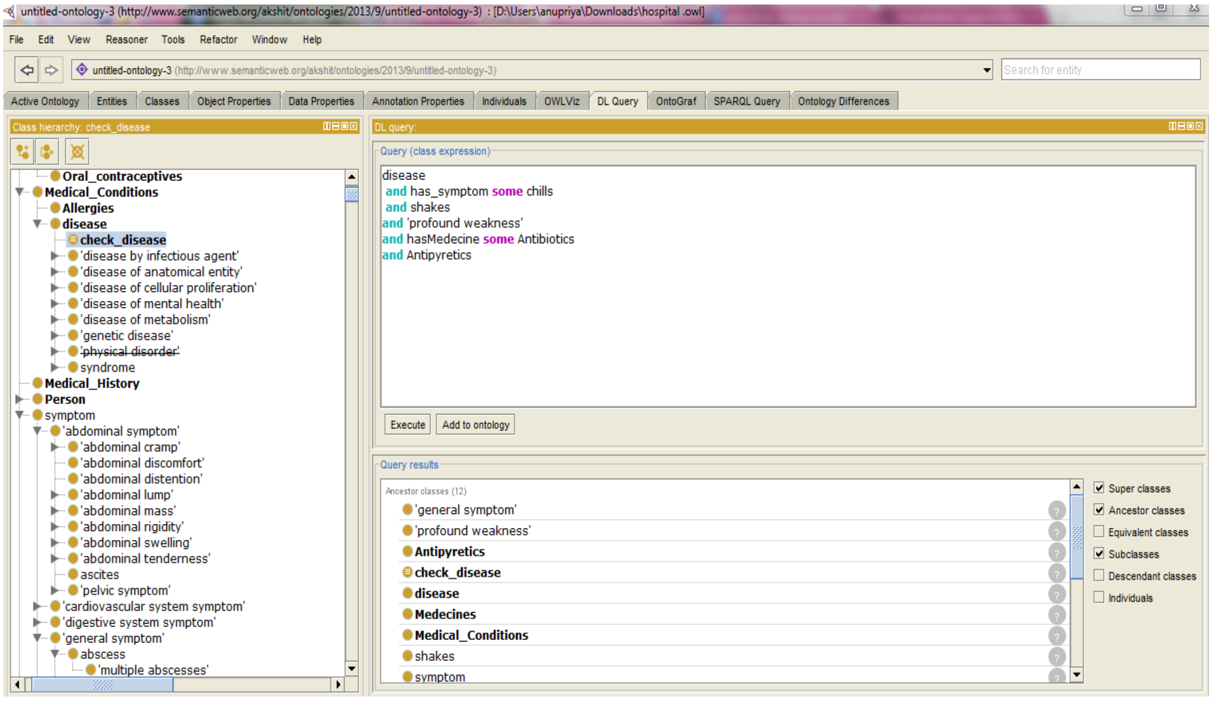
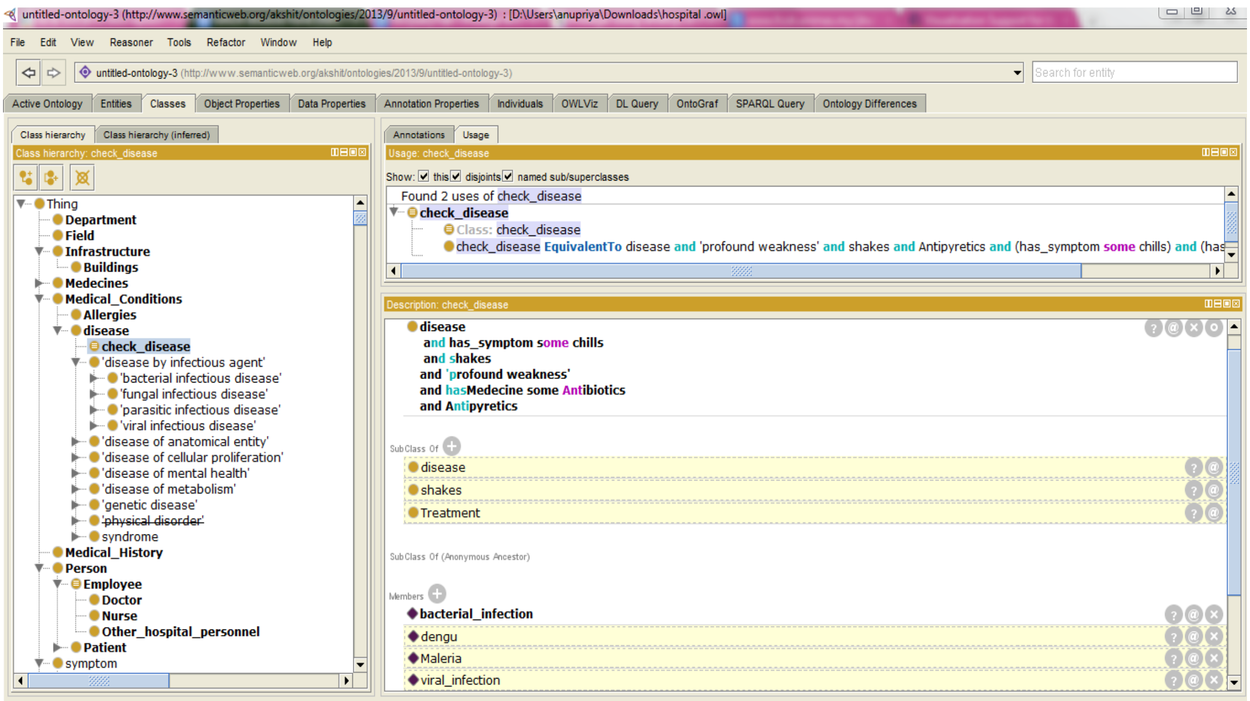


Figure 30 shows the results after execution of the *Reasoner*, the results were verified by the experts and were found to be consistent. In this case study Fact++ was used as the reasoner.

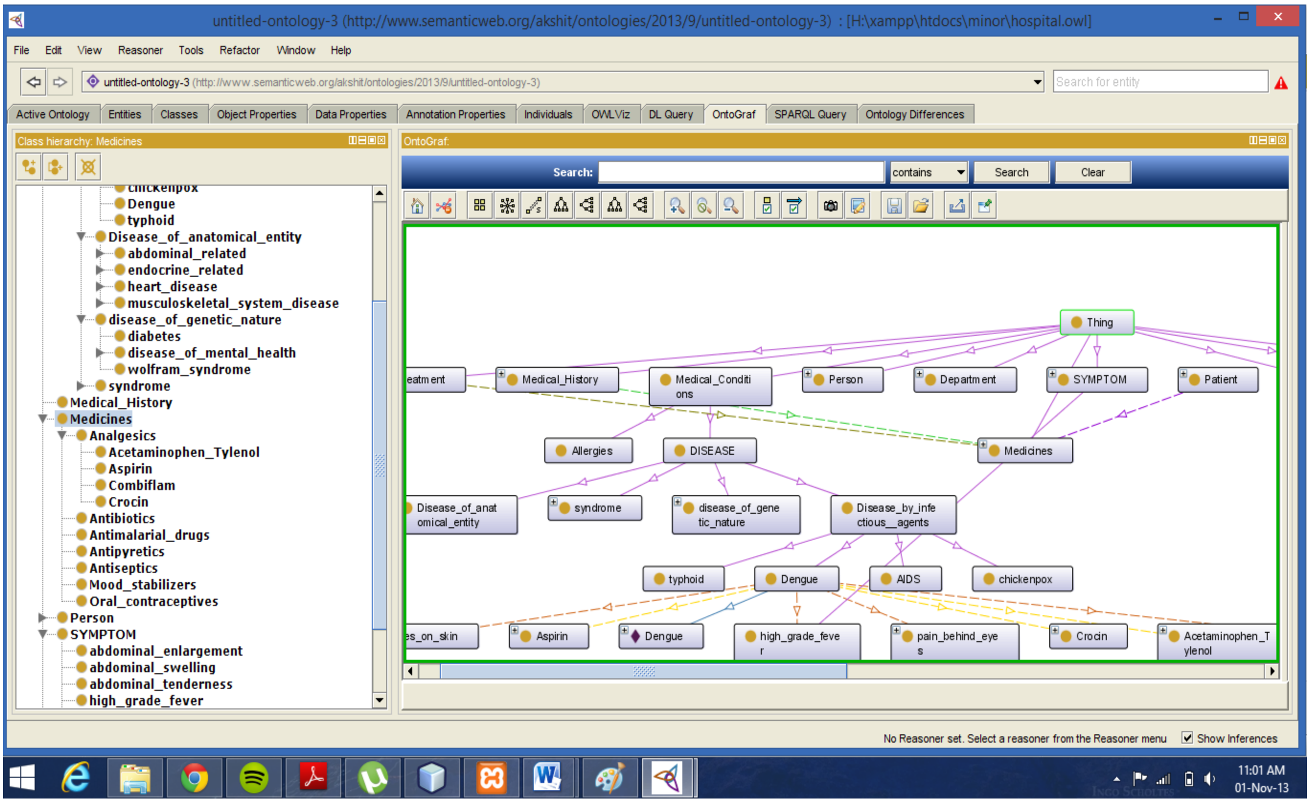
Fig 30. Evaluation of competency questions: after Reasoner execution



ii. Evaluation by OntoGraf and OWLViz

After executing the competency questions in the form of queries, it is required to check the correctness and completeness of the concepts and to verify the hierarchy and relationships given in the ontology. To perform this task OntoGraf and OWLViz provided by Protégé are used. Figure 31, 32(a) and 32(b) show the outputs of OntoGraf and OwlViz.

Fig 31. The output of OntoGraf



The relationships among the concepts are also verified by the experts using the directed links between the nodes in the OntoGraf.

Fig 32(a) A section of the output of OWLViz

Fig 32(b) A section of the output of OWLViz

6.2. Subjective evaluation of the enhanced On-To-Methodology

In this section, we present a comparison of the enhanced On-To-Methodology with other methodologies available for ontology construction. It can be seen from Table 6 that the On-To-Methodology contains many new features which were not previously defined in the most popular ontology construction methodologies. Due to space constraints and ease of understanding only the main ontology development methodologies have been considered in this comparison. There are many other recently developed methodologies such as the ontology construction based on the *Neon Approach* and *User Friendly Ontology Construction* based on use cases which are not included in this comparison. Table 6 presents the comparative analysis of different ontology development methodologies.

Metric for Evaluation	**Enhanced On-To-Methodology**	**METHONTOLOGY**	**Methodology by Farooq et. al.**	**MADRE**
Automation	Semi Automatic	Manual	Manual	Manual
Support for documentation	√	√	×	×
Steps defined in the Methodology				
Scope Identification	√ (Preliminary Specifications)	√	√	√
Knowledge Acquisition	√ (4 detailed sub steps defined)	√ (covered in single step, not defined in detail)	√ (covered in single step, not defined in detail)	√ (covered in single step, not defined in detail)
Ontology Design	√ (Detailed algorithm provided)	√ (Included in conceptualization and integration phase)	√	√ (Design Ontology Model)
Formalization/ Implementation	√/√	√/√	×/×	×/√
Evaluation	√ (illustrated with automation)	√ (no automation)	×	√ (no automation)
Verification/ Validation	√/√	√/×	√/√	√/×
Maintenance	√ (described in detail with flow charts)	×	×	×

Table 6. Mapping of concepts between enhanced On-To-Methodology & other methodologies

6.3. Automatic evaluation of the enhanced On-To-Methodology

In this section, the *hospital.owl* ontology, which is developed using On-To-Methodology is compared with a newly developed ontology named *HOSPnew.owl* (also referred as the HOSPnew ontology in this thesis), which is developed following the METHONTOLOGY method. The underlying idea is to prove the effectiveness of On-To-Methodology over METHONTOLOGY methodology using their case studies. The comparison is performed by calculating the ontology characteristics as defined by the IEEE standard named *Ontometric* [49], for both the ontologies. The ontology having better values for these characteristics is proved to be better and hence, the methodology used for its development is considered to be a better methodology.

This comparison can automatically be performed by the users through the developed tool On-To-Methodology Tool, using its *Evaluation Tab*. Sixteen ontology characteristics are chosen for evaluating both the ontologies, which are described as follows:

- *Methodology Followed*: Describes methodology used for construction of the ontology.
- *Implementation Language*: Describes language used for implementation of ontology.
- *Implementation Tool*: Describes tool used for implementation of ontology.
- *Number of concepts*: Presents the number of concepts defined in the ontology.
- *Number of sub concepts*: Presents the number of sub concepts defined in the ontology.
- *Number of Object Properties*: Presents the number of Object properties defined in the ontology.
- *Number of Data Properties*: Presents the number of Data properties defined in the ontology.
- *Instances Used*: Describes number of instances defined.
- *Depth of hierarchy of Ontology*: Presents the maximum depth of the class hierarchy.
- *Number of RDF links*: Identifies number of links used in RDF definition.
- *Number of Relationships*: Describes number of relationships defined.
- *Level of Inheritance*: Describes the level of inheritance used by the concepts.
- *Domains defined*: Provides the number of domains defined for constraints.
- *Ranges defined*: Provides the number of ranges defined for constraints.

- *Number of essential concepts*: It is the number of concepts in the highest level of hierarchy.

Figure 33 shows the snapshot of the class hierarchy of the ontology HOSPnew.owl. Figure 34 shows the OntoGraf evaluation of this ontology.

Fig 33. Class hierarchy of the HOSPnew ontology

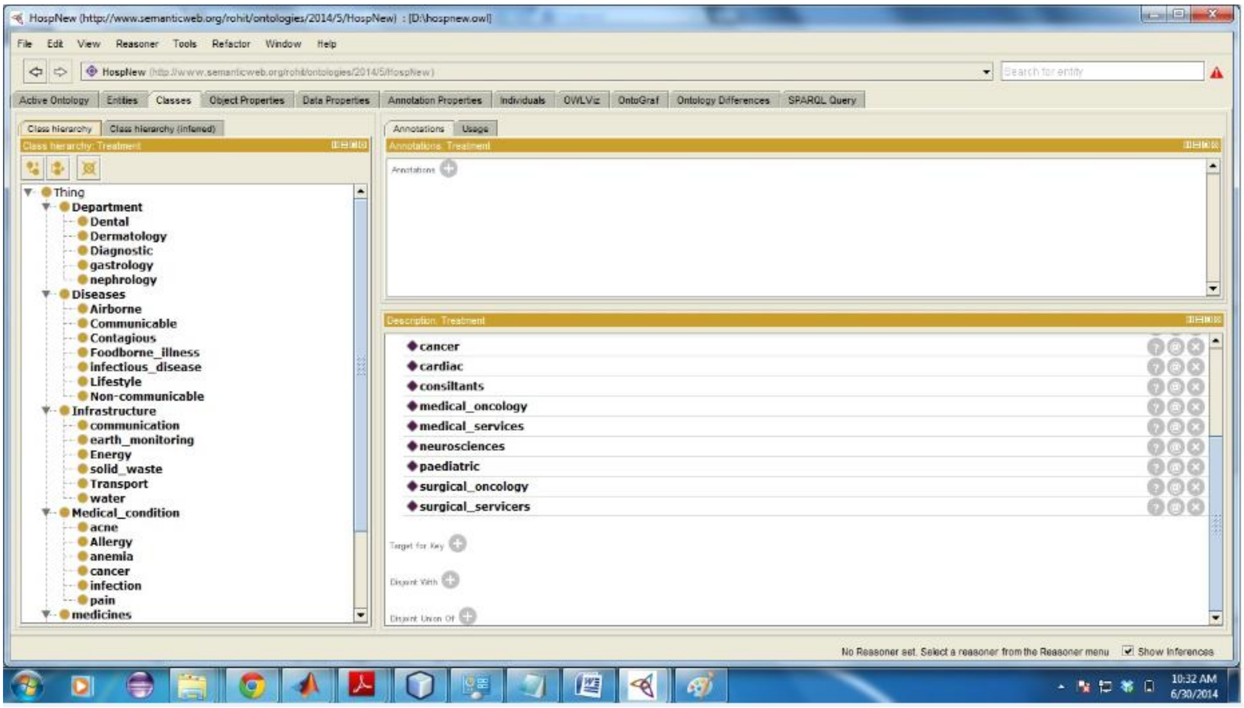
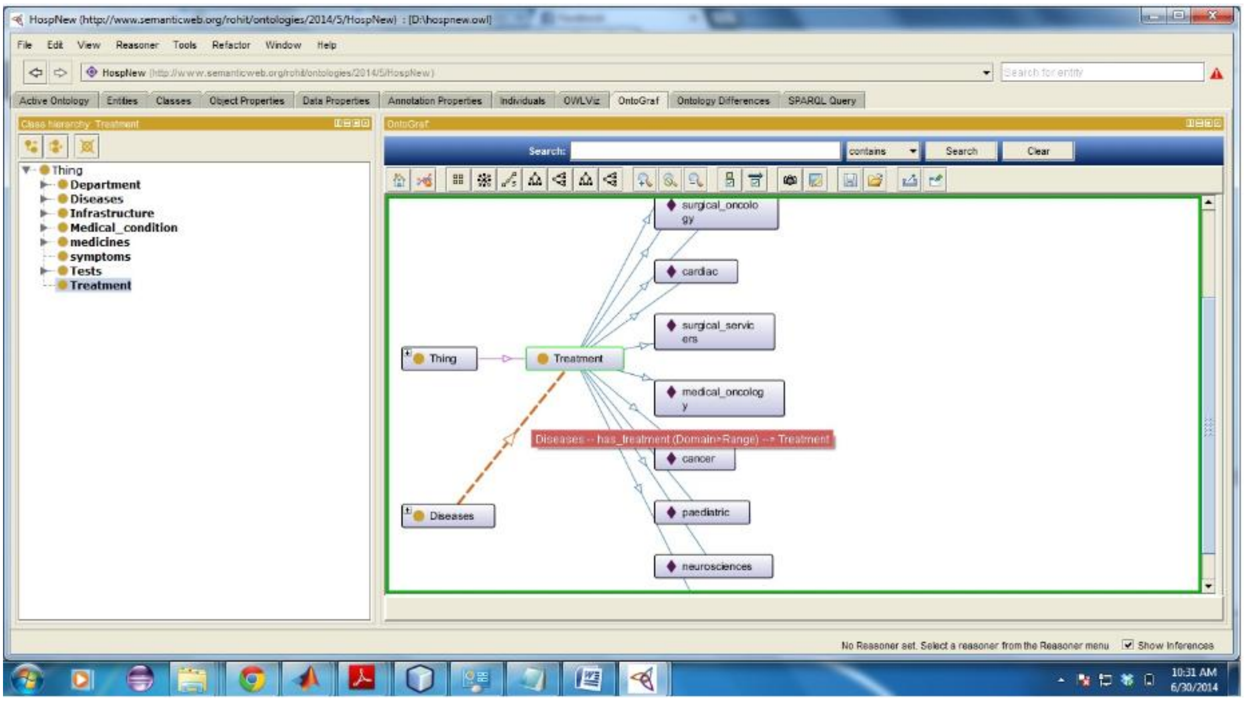


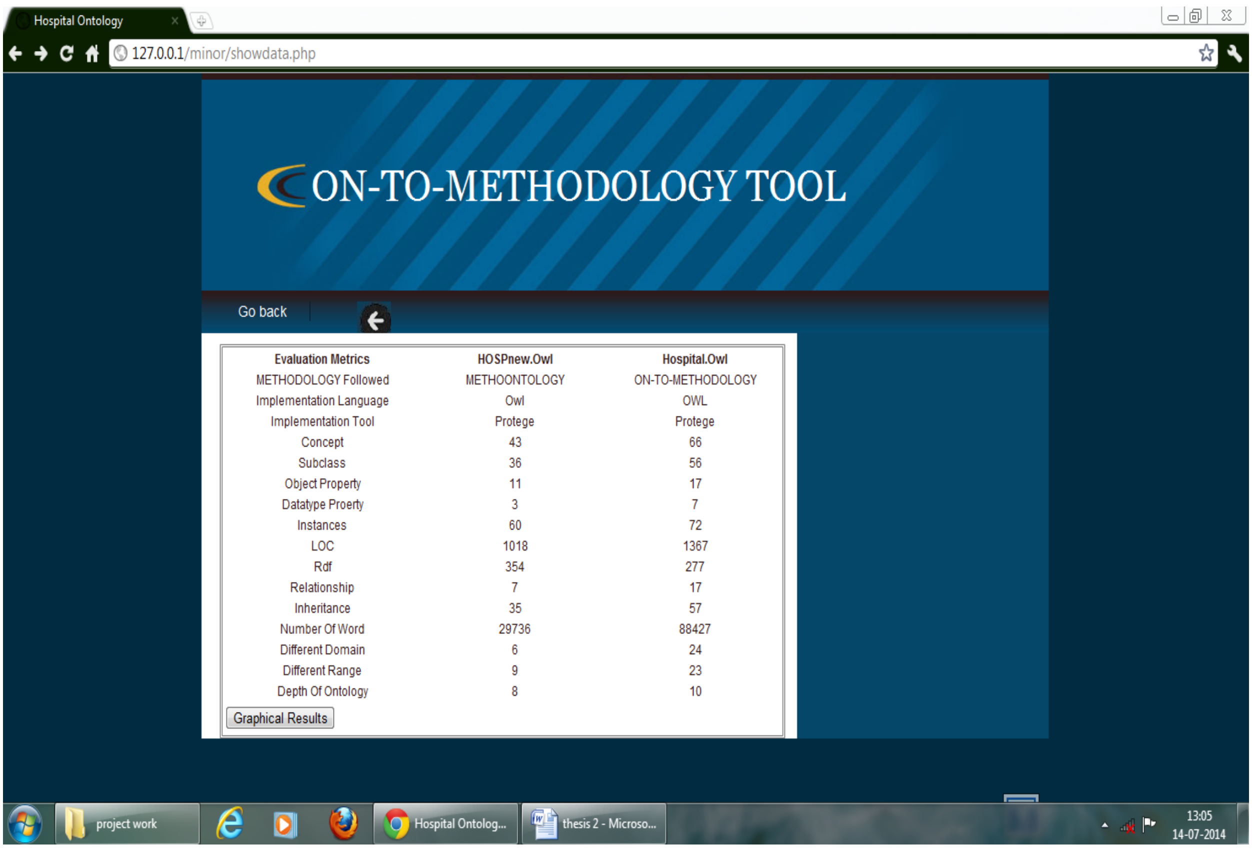
Fig 34. Output of OntoGraf for the HOSPnew Ontology



6.4. Results

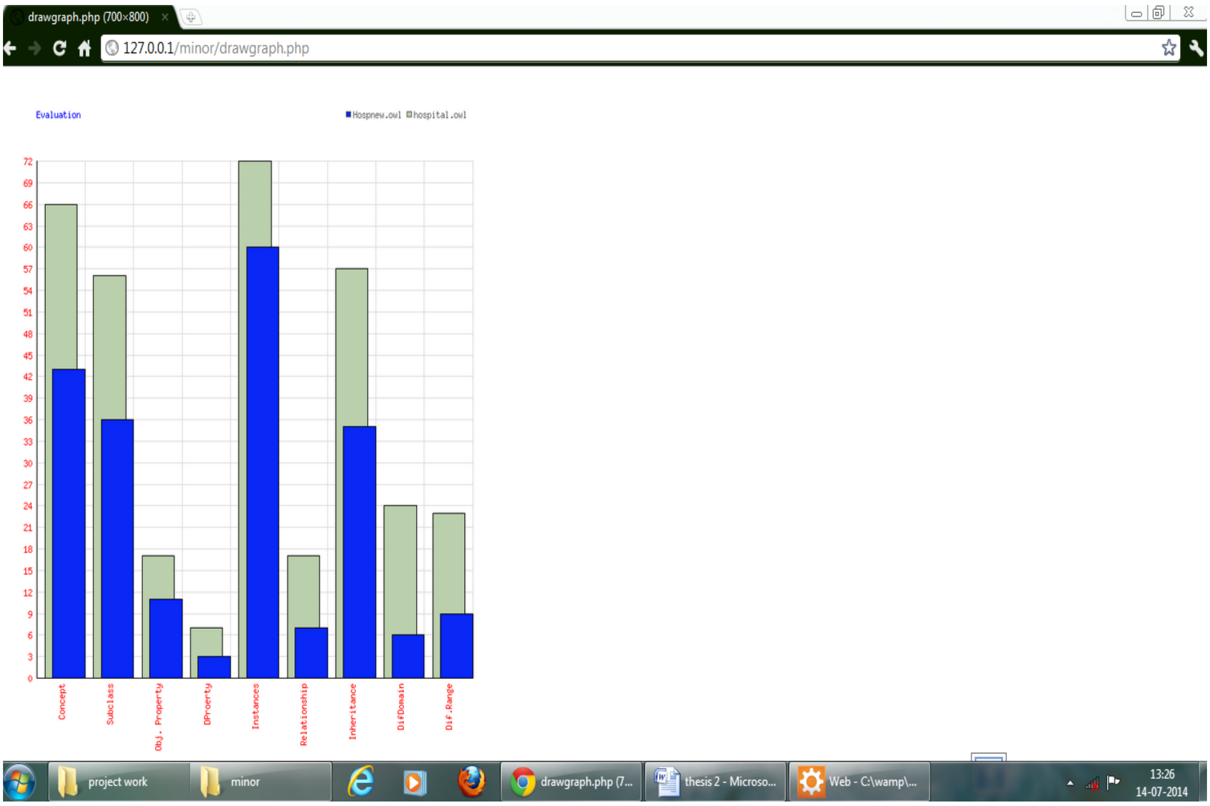
This section presents the results of evaluation of the proposed methodology. The evaluation was performed by comparing two ontologies developed for the same domain, using the proposed methodology and METHONTOLOGY respectively. Figure 35(a) and 35(b) show the snapshots of the results generated from the evaluation tab of the enhanced On-To-Methodology Tool.

Fig 35(a). Evaluation using the Tool



Evaluation Metrics	HOSPnew.Owl	Hospital.Owl
METHODOLOGY Followed	METHOONTOLOGY	ON-TO-METHODOLOGY
Implementation Language	Owl	OWL
Implementation Tool	Protege	Protege
Concept	43	66
Subclass	36	56
Object Property	11	17
Datatype Proerty	3	7
Instances	60	72
LOC	1018	1367
Rdf	354	277
Relationship	7	17
Inheritance	35	57
Number Of Word	29736	88427
Different Domain	6	24
Different Range	9	23
Depth Of Ontology	8	10

Fig 35(b). Evaluation using the Tool



The above graph shows the results of comparison of the two ontologies developed using On-To-Methodology and METHONTOLOGY methods respectively. The blue bars represent the results for the HOSPnew.owl file and the green bars represent the results for the hospital.owl file. The X- axis represents the metrics used for comparison and the Y-axis represents the values of the metrics. It is clear that the hospital.owl ontology is better than the HOSPnew.owl ontology file.

Table 7 shows the results concluded from the research. This table presents the comparative analysis of the all the three methodologies discussed in this thesis, namely, the enhanced On-To-Methodology, On-To Methodology proposed in [50] and METHONTOLOGY.

Comparison Criteria	Enhanced On-To-Methodology	On-To-Methodology by Magendra	METHONTOLOGY
Lifecycle Proposed	Iterative and Incremental	Sequential development	Evolving Prototypes
Case study domain	Healthcare domain	Bikes domain	Chemicals domain
Case study complexity	400+ concepts	Less than 50 concepts	284+ concepts, 103 instances
Availability of automation	Semi automatic	Semi automatic	Manual
Steps defined for ontology construction			
Step 1	Preliminary Specifications	Define Scope	Planning and Specifications
Step 2	Knowledge Acquisition • Any technique applicable • Use of both ontological and non- ontological resources	Knowledge Acquisition • Using FAST technique • Use of ontological resources only	Knowledge Acquisition • Using Interview technique • Use of ontological resources only
Step 3	Design Ontology • Using Jason classification and location map • Automatic	Ontology Design • Using location map • Automatic	Conceptualization • Using verb diagram and classification tree • Manual
Step 4	Implementation • Implementation language= OWL • Supporting tool= Protégé	Formalization • Formalization language= OWL • Supporting tool= Protégé	Integration
Step 5	Maintenance • Insertion of concepts or individual • Deletion of concept or sub ontology • Use of Jason string format	Evaluation	Implementation • Implementation language=Ontolingua • Supporting tool= Ontolingua Server

Step 6	Parallel Activities • Parallel Knowledge Acquisition • Integration • Documentation • Evaluation • Configuration Management	Maintenance • Insertion of concepts • Deletion of single concept	Evaluation
Step 7	×	×	Documentation

Prominent steps in development	Maintenance, Evaluation	Knowledge Acquisition, Design	Documentation
Evaluation Approach	Threefold Evaluation • Manual evaluation using expert knowledge • Semi automatic evaluation using protégé plug ins • Automatic evaluation using OntoMetric	Rule based Evaluation • Correctness Checking • Completeness Checking	Rule based Evaluation • Correctness Checking • Completeness Checking
Availability of automated evaluation	Automated	Semi automated	Manual

Table 7. Comparison of Enhanced On-To-Methodology, On-To-Methodology given in [50] and METHONTOLOGY

Chapter 7: CONCLUSIONS AND FUTURE WORK

This chapter presents the conclusions derived from this research and exhibits the possibilities of expansion of this work in future.

This thesis presents a new methodology for ontology development which is applicable to all domains, general in nature and easy to understand by novice users. It provides the researchers in the field of ontology engineering with a new way of constructing their ontology without worrying about its compatibility with other domains or the lifecycle of development. This work clearly presents the current ontology development state of art, issues in ontology development in the current time and tries to provide an unambiguous solution to these problems.

Since there exists no one methodology that can be applied in all scenarios of ontology development, our main emphasis here is on building a methodology that can be used as a standard method for ontology construction under various domains. This general methodology can be used with any life cycle and any application dependent or application independent environment. For this purpose, we have enhanced a proposed method named *On-To-Methodology*.

The methodology is also illustrated by working out Ontology for Hospitals. The methodology is provided automated support by developing the enhanced *On-To-Methodology Tool*. This tool not only provides support in ontology design and maintenance, but also provides automation for comparison of the proposed methodology with another popular methodology, named METHONTOLOGY. Moreover, the results of the comparison are presented to validate the efficiency of the enhanced *On-To-Methodology*. The results show that the otology developed using our methodology works better than the ontology developed using METHONTOLOGY.

We believe that this work can be improved and extended. The future work entails development of a tool that will automate the construction of conceptual model of the ontology. Further we can focus on developing a formal and unified method for eliciting the knowledge for ontology creation and automate the process of completeness and consistency checking in this phase.

Another dimension of future work is to focus on adding security concepts to the medical healthcare domain used as a case study in this research work, because lack of security features is one of the drawbacks of the developed case study. Any user of the system can make changes to the ontology as no security measures have been incorporated. Thus, it can be extended by adding security requirements to the medical healthcare ontology or automating the generation of security requirements from the given ontology by merging it with some security ontology.

Chapter 8: PUBLICATIONS FROM THE RESEARCH

This chapter briefly states the publications, both published and communicated, that have been worked out during this research work. The chapter is divided in two sections. The first section portrays the details of the published papers whereas the second section describes the details of the communicated papers.

8.1. Published Paper

This section briefs about the published research paper during the research.

1. Gupta D., Jaiswal S., Tewari A., ***"Security Requirements Engineering: Analysis and Prioritization",*** International conference on *Software Engineering Research and Practices, 2013, Las Vegas.*

8.2. Communicated Papers

This section presents the list of communicated research papers along with the details of the journals/conferences of publication.

1. Gupta D., Tewari A., *"**On-To-Methodology: Ontology Development Methodology",*** AI & Society Journal, vol 30.

2. Gupta D., Tewari A., *"**A Review and Classification of Security Ontologies",*** Artificial Intelligence Review, vol 42.

REFERENCES

[1] Guarino N., "Formal ontology in information systems",1998, IOS Pr.

[2] Borst P., H. Akkermans and J. Top, "Engineering ontologies", International Journal of Human Computer Studies, 1997, 46: p. 365-406.

[3] Anand S., Verma A., "Development of Ontology for Smart Hospital and Implementation using UML and RDF", International Journal of Computer Science Issues, Vol. 7, Issue 5, September 2010.

[4] Gruber T. R.., "A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition", 1993, 5(2): 199-220

[5] Gruber T. R.,"Towards principles for the design of ontologies used for knowledge sharing", International Workshop on Formal Ontology in Conceptual analysis and Knowledge Representation, The Netherlands, Padova, Italy,1993.

[6] Available Online at:
http://www.wikipedia.org/wiki/Ontology_(information_science)

[7] Gupta D., Jaiswal S., Tewari A., "*Security Requirements Engineering: Analysis and Prioritization",* International conference on Software Engineering Research and Practices, Las Vegas, 2013.

[8] Available Online at:
www.geneontology.org/

[9] Jabar M., Khalefa M., Abdullah R., Abdullah S., "Overview of types of Ontology in the software development process", IEEE Conference on Open Systems (ICOS), Sarawak, Malaysia, December 2013, p: 83-88.

[10] Monticolo D., Hilaire V., Koukam A., Gomes S., "Ontodesign; a domain ontology for building and exploiting project memories in mechanical design projects", Proceedings of Knowledge Management in Organizations, 2014.

[11] Evermann J., "The association construct in conceptual modelling – an analysis using the bunge ontological model", In proceedings of the 17th international conference on Advanced Information Systems Engineering, 2005, p: 33-47.

[12] Available Online at:
http://purl.bio**ontology**.org/**ontology**/EFO

[13] Rousseaux F., Lhoste K., “Towards a Collection-Based Knowledge Representation: the Example of Geopolitical Crisis Management”, From the book: Knowledge Management.

[14] Available Online at:
http://www.nlm.nih.gov/research/**umls**/

[15] M. Genesereth, R. Fikes, “Knowledge interchange format”, Technical Report Logic-92-1, Computer Science Department, Stanford University, 1992.

[16]M. Kifer, G. Lausen, J. Wu, “Logical foundations of object-oriented and frame-based languages”, Journal of the ACM 42 (4), 1995, 741–843.

[17] V.K. Chaudhri, A. Farquhar, R. Fikes, P.D. Karp, J.P. Rice, “Open Knowledge Base Connectivity 2.0.3”, Technical Report, 1998, available at:
http://www.ai.sri.com/_okbc/okbc-2-0-3.pdf

[18] I. Horrocks, F. van Harmelen, “Reference Description of the DAML+OIL”, Technical report, 2001, available at:
http://www.daml.org/2001/03/reference.html

[19] O. Lassila, R. Swick, “Resource description framework (RDF) model and syntax specification”, W3C Recommendation, 1999, available at:
http://www.w3.org/TR/REC-rdf-syntax/

[20] M. Dean, D. Connolly, F. van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, L.A. Stein, “OWL Web Ontology Language 1.0 Reference”, W3C Working Draft, 2002, available at:
http://www.w3.org/TR/owl-ref/

[21] D. Brickley, R.V. Guha, “RDF Vocabulary Description Language 1.0: RDF Schema”, W3C Working Draft, 2002, available at:
http://www.w3.org/TR/PR-rdf-schema

[22] Corcho O., Fernandez-Lopez M., Gomez-Perez A., “Methodologies, tools and languages for building ontologies. Where is their meeting point?”, Data & Knowledge Engineering 46, 2003, 41–64.9

[23] Gaoyun J, Jianliang W, Shaohua Y, “A Method for Consistent Ocean Ontology Construction”, IEEE 2nd International Conference on Industrial and Information Systems, 2010,37-42.

[24] Hepp M, “Possible Ontologies- How Reality Constrains the Development of Relevant Ontologies”, 2007.

[25] Deshpande N. et.al. "Construction and Applications of Ontology: Recent Trends", DESIDOC Journal of Library & Information Technology, Vol. 31, No. 2, March 2011.

[26] Lozano-Tello A., Gómez-Pérez A., "*ONTOMETRIC*: A Method to Choose the Appropriate Ontology", Journal of Database management, volume:15, p: 1-18, 2004.

[27] Protégé tool from Stanford University, available at: http://protege.stanford.edu/

[28] Fernández M, Gómez-Pérez A, Juristo N, "METHONTOLOGY: From Ontological Art Towards Ontological Engineering", AAAI Technical Report SS-97-06, 1997, available at: http://www.aaai.org/Papers/Symposia/Spring/1997/SS-97-06/SS97-06-005.pdf

[29] Fernández, M., "CHEMICALS: Ontology for chemicals", Final-Year Project, University of Madrid, 1996.

[30]Yang D., Tong L., Ye Y., Wu H., "Applying CommonKADS and Semantic Web Technologies to Ontology-Based E-Government Knowledge Systems", First Asian Semantic Web Conference, Beijing, China, September, 2006.

[31] Farooq A, Shah A, Asif K H, "Design of Ontology in Semantic Web Engineering Process", 2007.

[32] Zhang Y., Tan L., Liu J., Yu C., "A Domain Ontology Construction Method towards Healthy Housing", IEEE, Information Engineering and Computer Science, 2010, p:1-5.

[33] Lv K., Zeng Q., "A CPD-Ontology Built in IDEF5 for Product Development", Applied Mechanics and Materials, 2010, p: 268-271.

[34] Uschold M., King M., "Towards a Methodology for Building Ontologies", Workshop on Basic Ontological Issues in Knowledge Sharing, 1995.

[35] Grüninger, M., Fox M. S., Fernández López, "The Role of Competency Questions in Enterprise Engineering" IFIP WG 5.7 Workshop on Benchmarking. Theory and Practice. Trondheim, Norway, 1994.

[36] Grüninger, M., Fox M. S., "The Design and Evaluation of Ontologies for Enterprise Engineering", Workshop on Implemented Ontologies, European Conference on Artificial Intelligence, Amsterdam, NL,1994.

[37] Grüninger, M., Fox M. S., "Methodology for the Design and Evaluation of Ontologies", Workshop on Basic Ontological Issues in Knowledge Sharing, IGCAI, Montreal, 1995.

[38] Schreiber A., Wieling B.J., Jansweijer W.H., "The KACTUS View on the 'O' Word", Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI, Montreal, 1995.

[39] Bernaras, A., Laresgoiti, I., Corera, J., "Building and Reusing Ontologies for Electrical Network Applications", Proceedings of the European Conference on Artificial Intelligence ECAI, 1996.

[40] Swartout B., Ramesh P., Knight K., Russ T., "Toward Distributed Use of Large-Scale Ontologies", Symposium on Ontological Engineering of AAAI. Stanford, California, 1997.

[41] Dan Z, Li Z, Hao J, "Research on Semi-automatic Domain Ontology Construction", 2010 Seventh Web Information Systems and Applications Conference, p: 115-118.

[42] Yu Y., Hsu C., "A structured ontology construction by using data clustering and pattern tree mining", International Conference on Machine Learning and Cybernetics, Guilin, 2011, p: 45-50.

[43] Wei QUI, "Development and Application of Knowledge Engineering Based on Ontology", 2010 Third International Conference on Knowledge Discovery and Data Mining, p: 518-521.

[44] Chen Y J, Chen Y M, Huang C Y, Chao C Y, "Concept Feature-based Ontology Construction and Maintenance", 2010 IEEE, p: 28-32.

[45] Gupta D., Jat K., "Knowledge representation with Ontology and Relational database to RDF converter", Enterprise Information Systems and Web Technologies, 2008, p: 141-146.

[46] N. F. Noy, and D. L. McGuinness, "Ontology Development 101: A guide to creating your first ontology", Technical Report SMI-2001-0880, Stanford Medical Informatics, 2001.

[47] Tsarko D., Horrock I., "FaCT++ Description Logic Reasoner: System Description", available at:
http://owl.man.ac.uk/factplusplus

[48] Sirin E., Parsia B., Grau B., Kalyanpur A., Katz Y., "Pellet: A Practical OWL-DL Reasoner".

[49] Tello A., Gómez-Pérez A, "ONTOMETRIC: A method to choose the appropriate ontology", Journal of Database Management, 2004, volume: 15, pp: 1-18.

[50] Singh M., "Ontology development methodology: On-To-Methodology", M.Tech. Dissertation, Delhi Technological University, India, 2012.