Chapter 3

ON-TO-METHODOLOGY:
PROPOSED METHODOLOGY FOR ONTOLOGY CONSTRUCTION

In this chapter we shall describe our process of methodology. This process is generic and extension of proposed methodology by Fernández et. al. [21]. We create various documwnts at the end of each activity in the proposed methodology. These documents are analogous to various documents in software engineering. We call this methodology On-To-Methodology.

3.1 Process Description of Ontology Methodology

The different phases in On-To-Methodology are described in Figure 11. These steps are generic and consist of four main activities namely Knowledge Acquisition, Design Ontology, Formalization and Evaluation. The output of Knowledge Acquisition is Ontology Specification document. This document is analogous to Software Requirements Specification (SRS) document IEEE standard 830. The output of Design Ontology is Ontology Design Document (ODD) and consists of logical design of the ontology being developed. The output of Evaluation is Evaluation Report. It consists of the evaluation of ontology for knowledge representation efficiency.

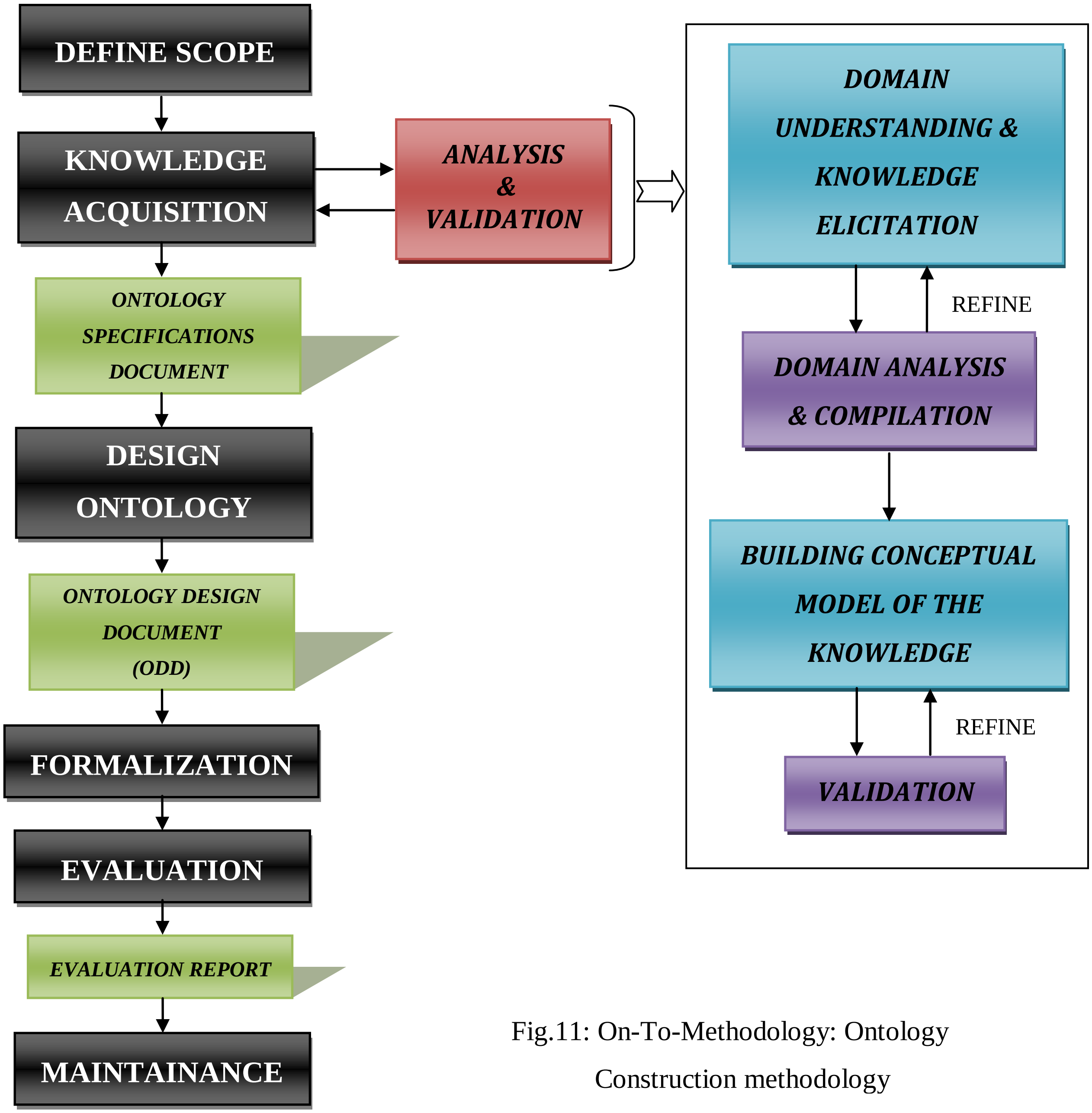


Fig.11: On-To-Methodology: Ontology Construction methodology

3.2 Define Ontology Scope

The target of this phase is to establish the scope of the ontology in terms of purpose that the ontology would serve with associated advantages. It should provide an objective description of the intended users, various scenarios in which it would be used and the end users.

3.3 Knowledge Acquisition

This activity would require the identification of *knowledge source* from where the knowledge is to be acquired. It can include the following:

- Books
- Experts
- Figures
- Tables
- Files
- Webpages
- Other ontologies etc.

Various techniques can be used for defining this such as brainstorming, interviews, group discussions, formal and informal reviews etc. The output of this phase is a *Ontology Specifications Document*.

We propose the use of *FAST* (Facilitated Application Specification Technique) here. It is a team oriented Software Engineering approach for gathering of requirements. This approach encourages the creation of a joint team of domain experts and users who work together to understand the expectations and propose a set of requirements.

We have adapted FAST to elicitate conceptual knowledge required to construct ontology in a formal way. We have designed various forms which are used to gather this knowledge. We identify the various stakeholders of the ontology and generate different forms to acquire the required knowledge. Figure 12 illustrates the various forms that we have generated.

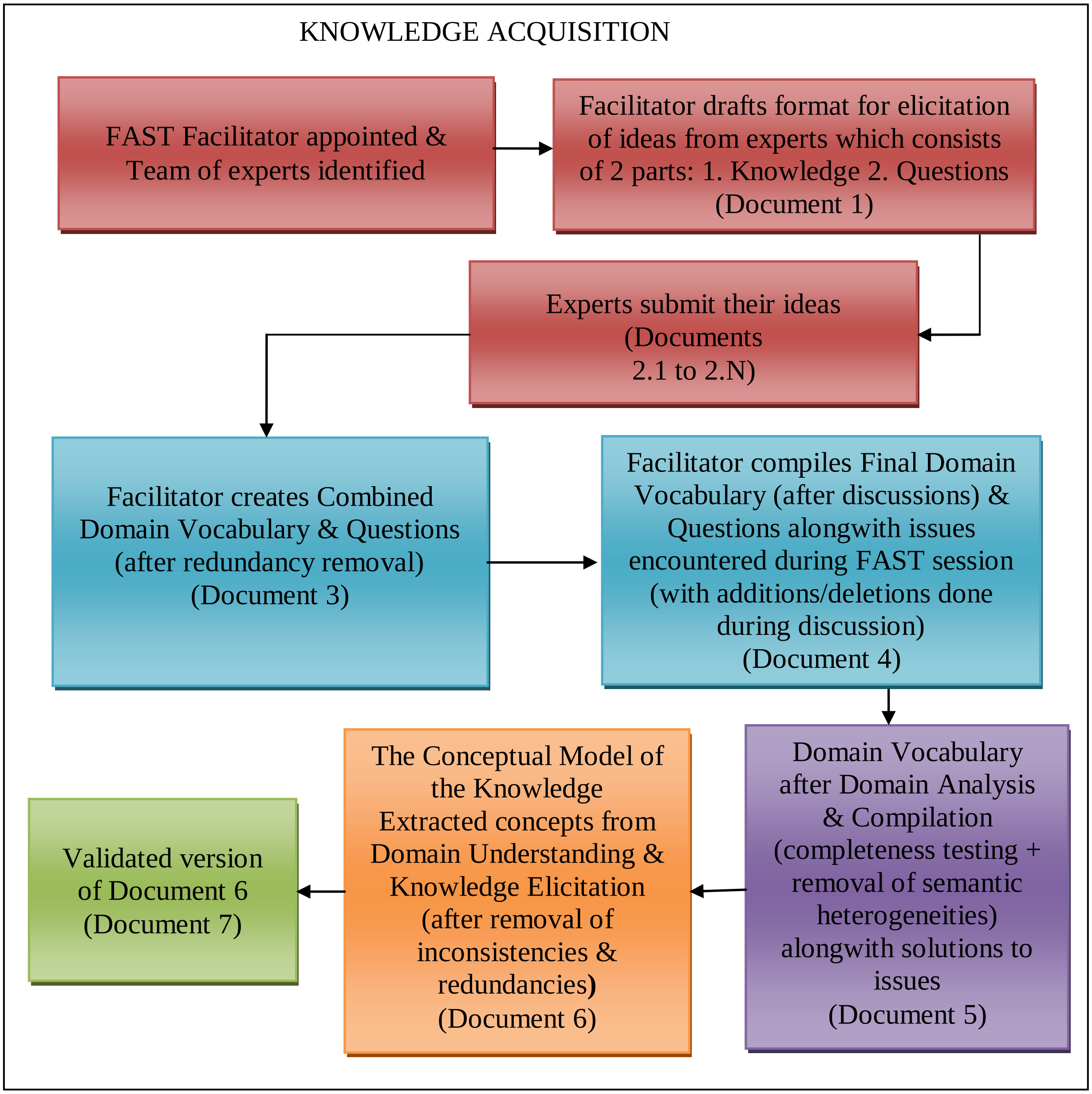


Figure 12: Various forms generated during different activities

As presented in figure, in the first phase a facilitator & team of experts are identified. The facilitator controls the various activities of the FAST session and prepares an informal agenda to encourage the free flow of ideas. The facilitator drafts a form (Document 1) to elicit ideas from the experts which consists of two parts which are required to be filled by the experts. The two parts are Knowledge and Questions. This form is filled by the various experts and submitted

Magendra Singh
M.Tech- Software Engineering
Delhi Technological University

back to the facilitator (Documents 2.1 to 2.N where N is the number of experts). The facilitator then compiles the ideas from these forms into a separate document (Document 3). The consolidated ideas are then discusses within the team and a final document i.e. document 4 is created which consists of the domain vocabulary and questions alongwith the issues encountered during the FAST session. The domain vocabulary listed in document 4 is analysed for the given domain and thus document 5 is created after compilatation. The concepts are then extracted from the Domain Understanding & Knowledge Elicitation as a result document 6 is obtained which is then validated and the final document consisting of the Conceptual model of the knowledge is obtained in the form of document 7.

We consider this activity consisting of four main tasks as described below. All these activities are carried out in a formal manner with the support of forms generated.

3.3.1 Domain Understanding & Knowledge Elicitation

Based on prelude

In this phase, the basic elements of the ontology are identified and enumerated. The elements of ontology are Class, Relationships, Constraints, Forms, Instances, Constants, and Instance attributes. These are explained as follows [28]:

- *Class* or concepts for a domain such as location, city, travel, destination etc.
- *Relationships* are the properties between two classes such as 'isa'.
- *Constraints* are conditions that must be satisfied during the design.
- *Instance* is values for particular categories in ontology.

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 elicitated in this phase. These questions are queries related to specific situations from real life problems that the ontology would help provide a solution. The competency questions are therefore used to evaluate the ontology thus developed.

3.3.2 Domain Analysis & Compilation

An expert for particular domain can easily examine the ontology in better way and changes suggested by expert will remove some of the inconsistencies at the earlier stage to make the design phase simpler. The expert review makes knowledge acquisition evolutionary [28].

So after drafting the basic model as described above as document 4, it is reviewed by the experts for following rules:

I. *Consistency rule*: various defects are analyzed for example usage of different names for a same concept.

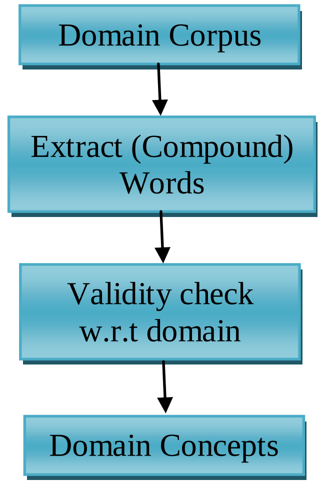
II. C*ompleteness rule*: the defects such as omission of domain resources and the omission of relationship are diagnosed. The issues encountered during FAST sessions are also resolved here.

If any consistency or completeness check fails, it is refined by the expert and the output is called as document 5.

3.3.3 Building Conceptual Model of the Knowledge

A conceptual model is developed on the basis of the domain model which we call as the document 5. Here the knowledge is represented in form of trees and table of rules/ formulas. The inconsistencies and the redundancies are resolved. The domain vocabulary identified in the *Domain Understanding & Knowledge Elicitation* is classified as classes, sub-classes, properties, constraints and individuals. The mechanism as shown in figure 13, can be employed.

Fig.:13 Framework for extracting the domain concepts



The definition of class hierarchy can be done in different ways [29] as following:

- Top-down development process begins with the definition of the concept of the field of the most common, followed by characterization of the concept.

- Bottom-up development process begins with the definition of the most specific classes, namely, class hierarchy the leaves, and then grouped into these classes more general concept.

- Combination of the two- defining more clearly the concept of the first, followed by their proper places generalization and characterization, it is the easier process.

The final result is document 6 we call as the conceptual model of the ontology.

3.3.4 Validation

The conceptual model of the knowledge, developed in the last step is reviewed here. Expert review is again used here. An expert for particular domain will examine the ontology for the following rules and the changes suggested will be incorporated [24]:

I. *Correctness rule*: The use of incorrect relationship, aggregation & specialization of classes and cardinality of relationship are tested.
II. *Rule for Semantic heterogeneities*: Semantic heterogeneities are also determined and dissolved in this step.
III. *Constraints*: Constraints on domain and range values of each object property and datatype property are also verified in the testing activity.

The result of this activity is document 7 which is the validated version of document 6.

3.3.5 The format for the Ontology Specification document is described as follows:

ONTOLOGY SPECIFICATION DOCUMENT FORMAT

Domain: ………………………………….
Date: ……………………………………..
Author: ………………………………......

1. Introduction

Analogous to IEEE std. 830-1998 for Software Requirement Specification (SRS), this document aims at defining the overall requirements for Ontology being developed. The final ontology will be having only features/functionalities mentioned in this document and assumptions for any additional functionality/feature should not be made by any of the parties involved in developing/testing/implementing/using this product. In case it is required to have some additional features, a formal change request will need to be raised 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 of the ontology. This document is also the starting point for design phase of ontology development methodology and is also used for testing the ontology when developed.

1.2. Scope

Here the scope of the ontology is established in terms of purpose that the ontology would serve with associated advantages. An objective description of the intended users is also provided here alongwith the various scenarios in which it would be used and the end users.

1.3. Definitions, acronyms & abbreviations

Here the various terms are defined and all the acronyms & abbreviations using during the documentation are listed.

1.4. References

A complete list of all the referenced documents is provided here.

1.5. Sources of knowledge

The various sources of knowledge from where the elements of the ontology are derived are listed under this sub-section.

1.6. Overview

The contents & organization of the rest of the Ontology specification document is described here.

2. **Overall Description**

The general factors that affect the ontology are described under this section.

2.1. Ontology Functions

The ontology will store the following elements:

- Class hierarchy
- Properties
- Inverse properties
- Instances

Based on the above information, the ontology will help solve some problem at hand. The major functions of such a system are listed here

2.2. User Characteristics

The characteristics of the intended users are listed here with the following details:

- Educational Qualification:
  The required educational qualification that the user should have to use the system is provided here.

- Experience Requirements:
  Any experience requirements of the user are provided here.

- Technical Expertise:
  The technical expertise that the user should possess is provided here.

2.3. Constraints

An overview of the constraints applicable to the ontology system is provided here.

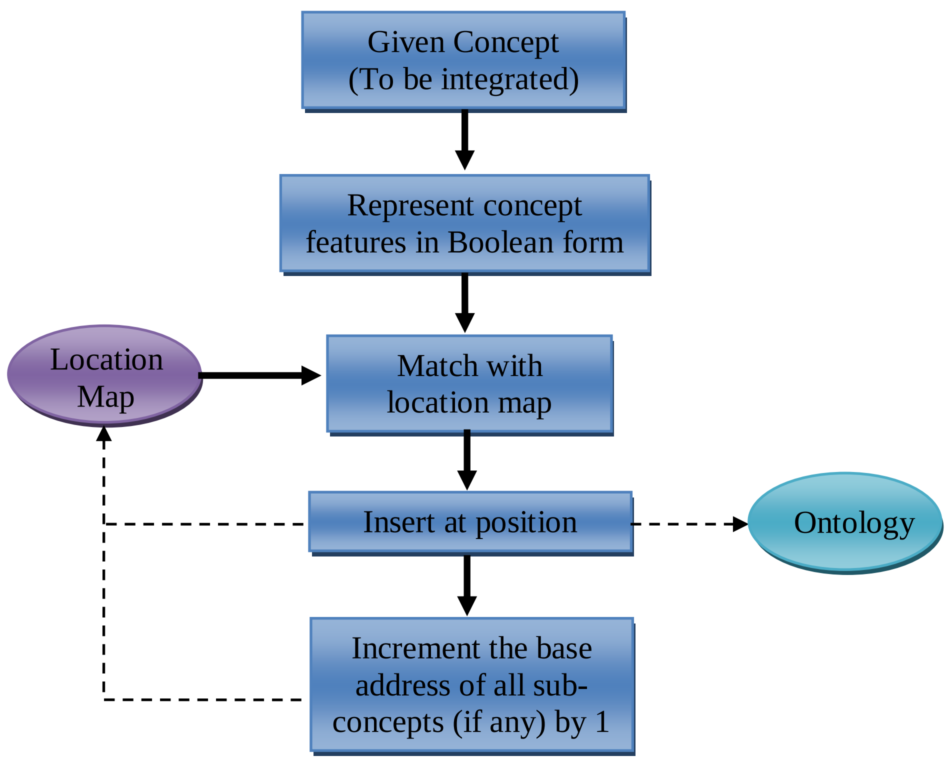
3.4 Design Ontology

This phase consists of two main tasks. One is the design task where with the help of the Ontology Specification Document and the Conceptual model of the knowledge, the concepts are mapped to location in location map. In this phase the concepts in the conceptual model are structured so that knowledge can be accessed. Second task is to produce a design document called Ontology Design Document (ODD) which can be helpful in evolution of the ontology.

3.4.1 Design

This is the core task of building ontology where different concepts such as classes, instances etc. in the conceptual model are processed to form the structure of ontology. We have adapted an algorithm for this and apply the technique [15] as shown in figure 14 below:

Fig.14: Adding new concept



The steps as shown in figure 15 can be explained as follows:

i. A concept is selected from the conceptual; model.
ii. It is then converted to Boolean form by representing its features in Boolean form.
iii. The concept as represented in above step is then matched with other concepts already present in location map.
iv. On determination of the correct position of the concept in the ontology structure (/location map), the concept is inserted in that position.
v. If there exist any sub-concepts concepts then increment their base address by 1.

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

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 say 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 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 is not present in X, then X is not a sub-concept of Y thus it is either a brother concept of this concept or a sub-concept of some 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.

The following are the snapshots of our tool:

Figure 15: Home screen

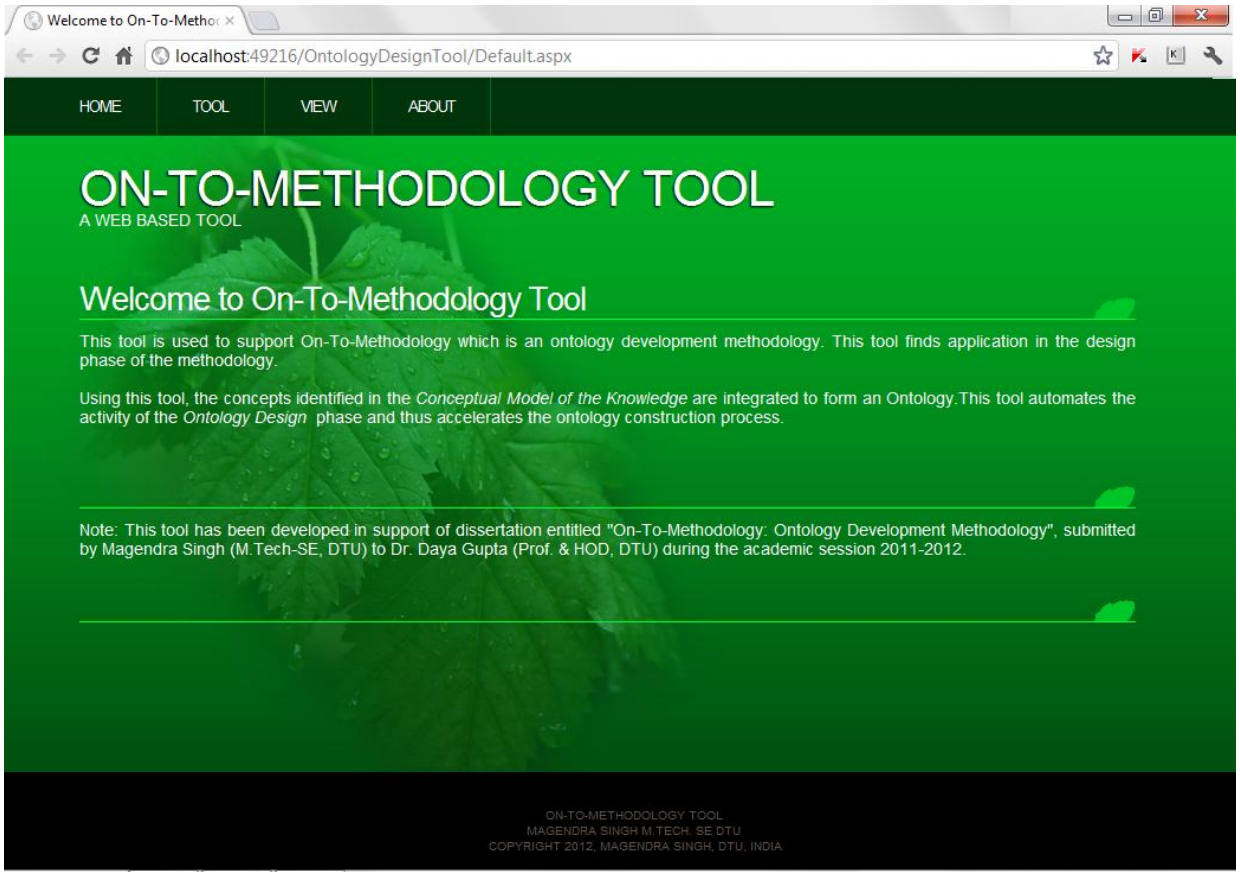


Fig.16 (a): Interface for designing

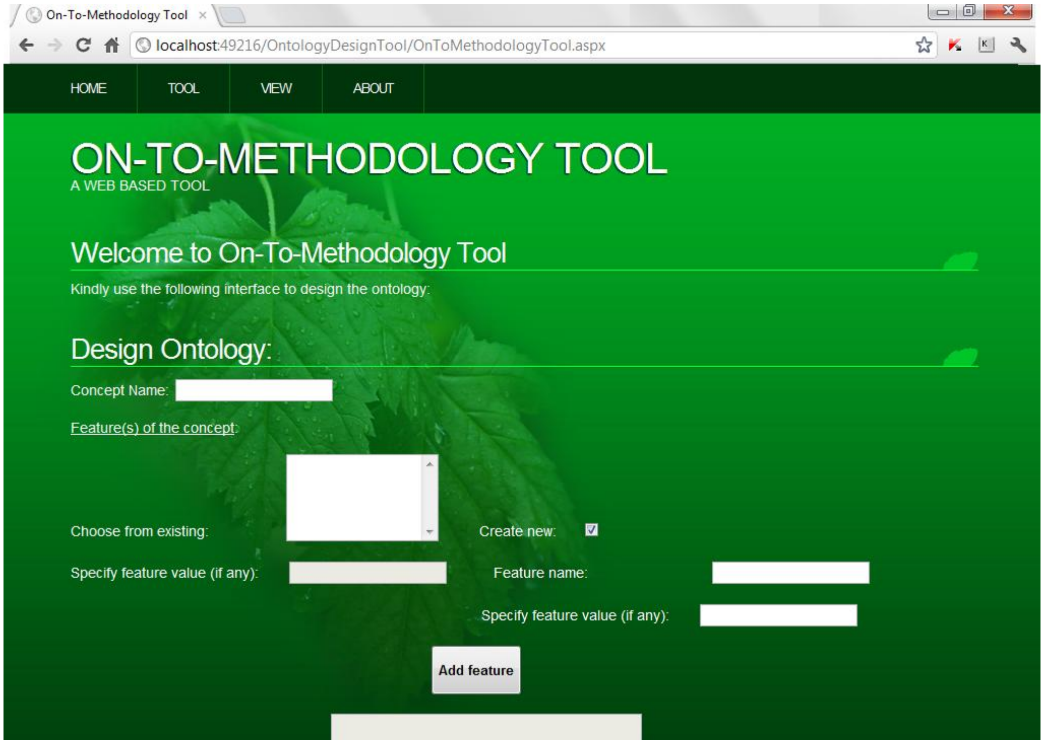
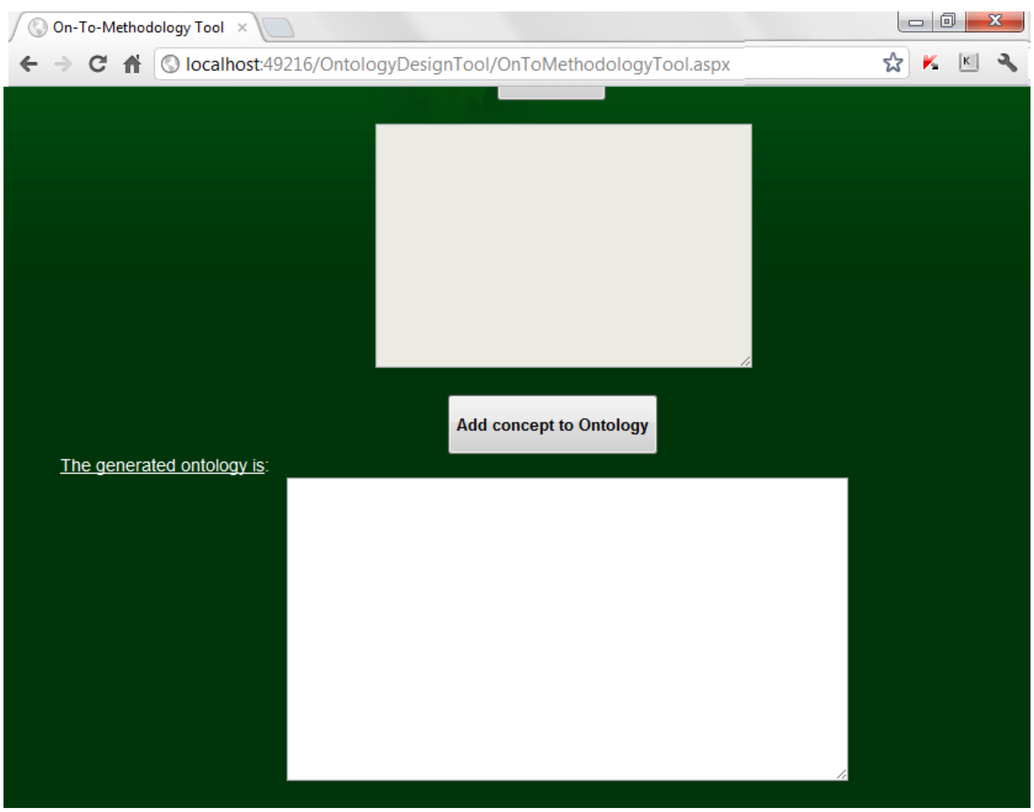


Fig.16 (b): Interface for designing



3.4.2 Ontology Design Document

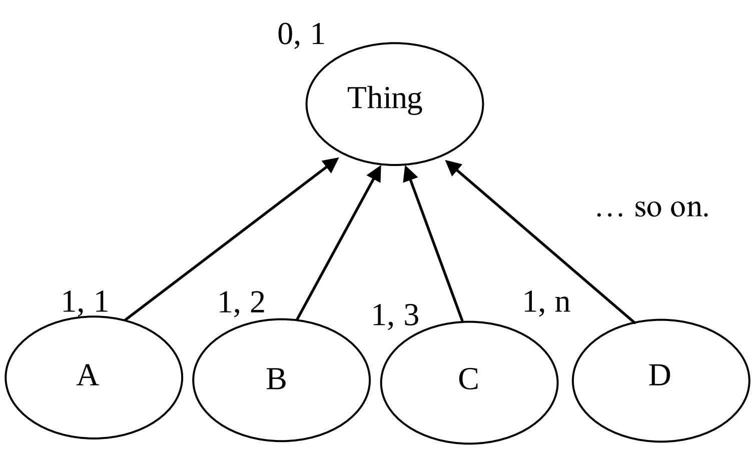
The output of this phase is the Ontology Design Document (ODD). This document is a formal record of the ontology structure and consists of the Location map alongwith the graphical representation of the ontology structure. The format for ODD is as follows:

ONTOLOGY DESIGN DOCUMENT

The location map is as below:

CONCEPT ADDRESS	CONCEPT FEATURE
Thing (0, 1)	X
A (a1, a2)	$f^{+}1 \bullet f^{+}2 \bullet \ldots$
B (b1, b2)	$f^{+}x \bullet f^{+}y \bullet \ldots$
…	…

The graphical representation of ontology structure is as follows:



3.5 Formalization

The main aim behind ontology development is to make the information to be understood by the machines. Thus, the ontology is finally formalized using the selected ontology languages and tools.

Semantic Web is not a single technology and it comprises of a number of components including ontology languages (RDF/RDFS, OWL etc.), editing tools (Protégé, Ontolingua etc.) and standards (WSMO, OWL-S etc.). As quoted in [28], different ontology languages provide different facilities. The most recent development in standard ontology languages is OWL from the W3C. OWL [8] makes it possible to describe concepts but it also provides new facilities. It has a richer set of operators - e.g. and, or and negation. It is based on a different logical model which makes it possible for concepts to be defined as well as described. Complex concepts can therefore be built up in definitions out of simpler concepts. OWL ontologies may be categorized into three sub languages: OWL-Lite, OWL-DL and OWL-Full. A defining feature of each sub-language is its expressiveness. OWL-Lite is the least expressive sub-language. OWL-Full is the most expressive sub-language. The expressiveness of OWL-DL falls between that of OWL-Lite and OWL-Full. Islam et. al. [30] have compared several editing tools as shown in Table 1 below.

Table 1: A comparison of ontology editing tools

Tools	Free	Open Source	Dot Net and Web Based	Import Languages	Export Languages	Inferencing Reasoning Tools
Protégé	√	√	×	RDF, OWL	RDF, OWL, FLogic, CLIPS	√
OntoEdit (Free)	√	×	×	DAML + OIL, RDFS	DAML + OIL, RDFS	×
Differential Ontology Editor (DOE)	√	×	×	RDFS, OWL	DAML+OIL, OIL, RDFS, OWL	×
IsaViz	√	√	×	RDF/XML, N-Triples	RDF/XML, N-Triples	√
Ontolingua	√	×	×	DAML+OIL, KIF,CLIPS	DAML+OIL, KIF, CLIPS	√
Altova Semantic Works™	×	×	×	N-triples, OWL, RDF and RDFS	N-triples, OWL, RDF and RDFS	×
WebODE	√	×	×	RDF(S), DAML+OIL, UML, OWL	CLIPS, DAML+OIL, UML, OWL	√
TopBraid Composer	×	×	×	RDBMS, OWL, RDF	OWL,RDF, XML	√
Morla	√	√	×	RDF, OWL	RDF,OWL	×
Hozo	√	√	×	RDF, OWL (subset)	OWL, RDF	×
TODE	√	×	√	RDF, OWL-Lite, N-3, RDBMS, N-Triple,	RDF, OWL-Lite, N-Triple, N-3, RDBMS	×

3.6 Evaluation

After the ontology has been drafted, it should be checked for quality and knowledge representation efficiency. If any constraints are violated then they are corrected. The ontology engineers also verify the developed ontology against the requirement document.

The term 'Evaluation' encompasses both- verification & validation. Based on [25], we identify following two ways of evaluation:

- *Relation Evaluation*: Relation evaluation infers the new individual based on constructed rule set, and judges whether the relation is coincident with the professionals' knowledge. This is done using the *Reasoner* that comes inbuilt in the tool, like Fact++ in Protégé.

- *Evaluation of hierarchy*: Evaluation of hierarchy develops limits definition of created class and individual properties, 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.

Also, in above steps, the answers to competency questions are verified by the experts. The output of this phase is the *Evaluation Report*. It consists of following:

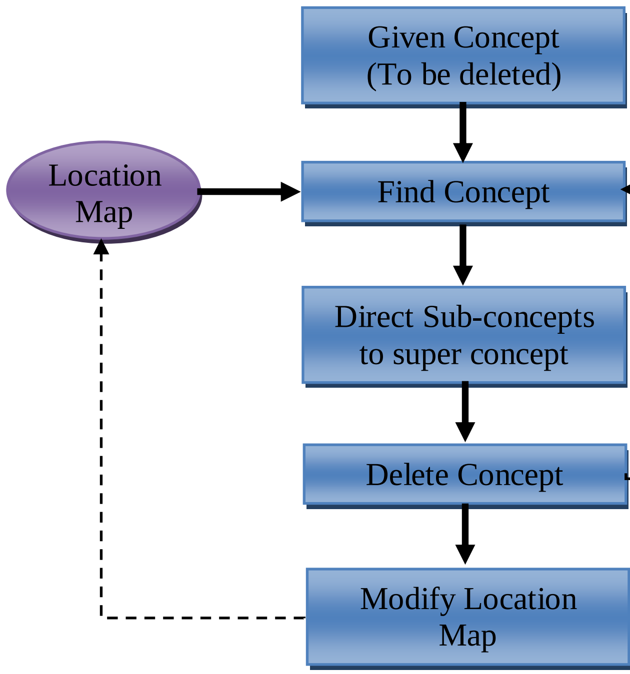
I. Evaluation of competency questions by the experts to check if the results are as per the expectation.
II. Evaluation of hierarchy using OntoGraf & OWLViz.

3.7 Maintenance

Ontologies need to adapt to the changing specifications to become more mature. This can be accomplished by adding, updating and removing concepts/ parts of the ontology. For this, following techniques can be employed:

- Addition of new concept: Technique as used in ontology design can be used.
- Deletion of a single concept: Technique as shown in figure 17 can be used.

Fig.17: Deletion process of single concept



- Deletion Operation for a Portion of the Ontology: Technique as shown in figure 18 can be used.

Fig.18: Deletion process of a portion of ontology