3. DOMAIN KNOWLEDGE MODELLING

3.1 INTRODUCTION

To design a methodology for creating the ontology in the domain of Computer Science (CS) and also develop a personalized ontology for each user by collecting the information has been one of the primary objectives of this research work. The development of ontology has been done by taking into account the advantages of conceptualizing domain knowledge and also the scale of formality in terms of logical theories, taxonomies and metadata schemes. The ontologies are basically the content theories and the main contribution of ontologies is to recognize the classes, objects and the relationships in a particular knowledge domain. Ontologies provide a potential vocabulary for unfolding the knowledge about a particular domain. Ontologies will help a learner in course planning as it displayed the full course domain in a hierarchy with the relationship among the courses and there are cases where inter-domain courses will be shown linked together. The ontology developed for CS will support in increasing the number of recommending courses to the learners’ after the courses are mined from the dataset. The need is to provide appropriate courses to the user based on the area of interest. The advantages of ontology study are as follows.

- To provide visualization of the computer science domain
- To share a common understanding of computer science theory
- To describe the terminology of computer science knowledge model by explaining the concepts, Instances and the properties

3.2 ONTOLOGY CLASSIFICATION

Ontology is described as a formal, unambiguous, and explicit description of a shared conceptualization in a domain of discourse as shown in Figure 3.1. The Ontologies were applied to extract information and knowledge from various domains, but in the domain of CS, it is deficient and restricted to certain topics.
The basic understanding of a domain will be provided by the domain ontologies which are shareable and include the concept and the relationship between them. The common points of interest in approximately all kinds of ontologies are classes because a class would define a concept about a particular domain. Domain ontologies together with a group of individual classes would represent a knowledge base. The ontologies are considered a well-organized methodology for knowledge modelling, which are intended to create knowledge representation besides classifying a particular domain which could be shared and reused [82]. The construction of 5-tuple ontology is outlined as $O = \{C, R, I, H_c, H_r\}$. The construction of 5-tuple ontology is outlined as $O = \{C, R, I, H_c, H_r\}$. The construction of 5-tuple ontology is outlined as $O = \{C, R, I, H_c, H_r\}$. The construction of 5-tuple ontology is outlined as $O = \{C, R, I, H_c, H_r\}$. The construction of 5-tuple ontology is outlined as $O = \{C, R, I, H_c, H_r\}$. The construction of 5-tuple ontology is outlined as $O = \{C, R, I, H_c, H_r\}$.

$$O = \{C, R, A\}$$

Where $C$ is a concept, $A$ is a set of axioms on $O$ and $R$ is Relationship.
Here $C$ represents a set of concepts, $I$ represent instances of concepts (classes), $R$ represents a relationship set between two concepts $r = (C_1, C_2) \in R$ or $R (C_1) = (C_2)$ and $Hc$ represents the hierarchy of the sorted concepts, where $R_i \in R$ and $R_i \rightarrow C \cdot C$. $H_r$ and $H_c$ depict hierarchies of relations and concepts respectively [83]. The building blocks for all ontologies are classes as it describes the concept in the domain and that is the reason it is considered an organized methodology for knowledge modelling. Ontology should closely match the level of information found in textbooks and curriculum on a particular subject and model the related information that is known about that subject. Thus, for building an ontology following kind of concepts need to be specified as.

- Classes and objects present in the domain
- Properties those objects would exhibit
- Relationship that exists between objects

Ontologies are categorized based on two aspects, the concept of internal structure and the kind of information being captured. In CS, the classification of ontology is done by objectives and conceptualization of subjects such as application ontology, domain ontology, domain-task ontology, and task ontology presented in Figure 3.2.

![Figure 3.2 Classification of Ontology](image)

The application ontologies encapsulate the knowledge pertinent for a particular application and, as a result, these ontologies are application reliant. The application ontology could expand their domain with a great deal of relevance with respect to a specific problem or application. The task ontologies are limited
to a specific activity or a domain and correspond to a particular vocabulary for some task within a specific application. The task ontologies can’t be used outside its working environment or from one domain to another. Instead, domain ontologies which are used within a particular domain but they possess reusability feature e.g. medical, history, sports and entertainment ontologies.

They main feature of all domain ontology is their autonomous nature in a particular application and they also provide a vocabulary to a concept and represent the knowledge in a particular domain. Task ontology on the other hand defines the related terms with respect to a normal task such as selling and buying are domain independent tasks. This study employed domain ontology for the computer science subjects and the key elements of an ontology are showed in Figure 3.3

![Figure 3.3 Ontology Components](image)

**3.3 ONTOLOGY DEVELOPMENT TOOLS**

Since the concept of ontology came into practice, huge number of software tools has been developed to create and maintain the ontology. The prominent among the tolls are Protege, OntoLingua, KAON, WebOnto, OntoEdit, and RDFedt. In this study, the Protege tool has been utilized for the development of computer science ontology.
3.3.1 Protege

Protege is a free source knowledge based framework available which provides support to OWL language and Resource Description Framework (RDF) specifications. Protege is helpful to build ontology, visualizing ontology and ontology editing using frames. It is a java based tool which save ontologies in various file extensions, support plug-in like OntoGraf and OntoViz for visualizing ontology and can be exported into different formats that include RDF, OWL, N3, CLIPS, Turtle, and some common metadata language description of web such as RDF and RDF Schema [84].

The applications based on ontology are the key constituents for the web semantics as it primarily focuses on the concepts and the relationships rather than information alone. In the present scenario, the owl ontology construction serves as the foundation for semantic network technology that is being created by protege and ontology is known as owl document [85]. Protege interface is intuitive and simple to use but at the same time, it offers lots of advanced options for working on ontology development. Other possible ontology editors that are available to use like Chimeara, OntoLingua are all web-based tools for ontology development and needs a login to servers to access. However, due to lack of facility to access these tools Protégé was chosen as it was easily available.

Protege is having a flexible GUI for creating a knowledge model and also has a plug-in feature that is more apt for the ontology developers for concentrating on concepts instead of the output language syntax. Present study has added a few significant domain classes along with related subclasses of courses of BSA Crescent Institute of Science and Technology. Another plug-in the protege has facilitated for visual assistance is OntoGraf and OWLViz. Using these visualization options the assorted and the inferred domain models of the developed ontology can be seen [86].

Furthermore, protege has a built-in reasoner as an engine of inference to check the inconsistencies on the ontology classes which mainly focus on the errors and relationships. The finest reasoner available that is designed to work
with the current ontology tool and is written in OWL format is Fact++ and Hermit. Reasoner is vital for developing the consistent and correct ontology. The input to the reasoner is ontology and tries to provide output in the form of relation and statements. Reasoner finds the logic based inconsistency in classes and relationships and later highlights all those classes that contradict. In current research FACT++, reasoner was chosen to check any inconsistency and some Description Language Query (DLQ) were used to do some other tests. To visualize the ontology protege use a different kind of methodology which is in the form of node links, indented lists and tree view etc

Protégé comes with lots of tabs among them the mostly used tabs are Class, Properties, Individual, Metadata and Form Tabs. The Class tab as its name suggested is used to delineate and their hierarchy. There are three types of property tabs Annotation, Object and Data property tabs and all these property tabs define the function of the property, its domain and the range of the properties.

3.4 ONTOLOGY DEVELOPMENT LANGUAGES

There are various languages available to represent the knowledge of domains such as Engineering, Medical, Academics and Science field. This section will give a brief about various representation languages and about OWL which is being widely accepted and used for knowledge representation of the domain of interest. During the past decade, various languages have been built for ontology developing, implementation and maintenance such as OKBC, Carin, FLogic, OntoLingua, OCML, ApolloLoom, TELOS, and Cycl [87]. All the mentioned languages have precise ontologies because they are conventional and consist plain text only.

As technologies got advanced other languages on web ontologies were developed such as XOL, OML, OIL, RDF, and OWL. All of these languages are based on XML syntax and has been already adopted as a benchmark. These ontology languages allow the developers to encode the knowledge about a particular domain which includes the rules of reasoning to support automatic
processing of knowledge with less or zero human intervention. These languages are very much closer to First Order Logic (FOL) or reasoning due to which they are more effective than simple data storage model. Another important feature of ontology is in its nature of interoperability because of which it can be reused or shared among machines and humans. Ontology can be used for heterogeneous database integration among different systems. This is the reason that ontologies are considered as basic ingredient for semantics whereas databases are simply a schema model.

3.4.1 Classification of Ontology Languages

The ontology can be represented by various ontology representation languages developed over a period. Some of the languages use XML schema to code knowledge, while some use languages based on frames and extension of frames called Description Logic (DL) [88]. The description logics are considered more suitable for ontology development because of high quality ontology is crucial for the semantic web. The Ontology languages are classified as given in Table 3.1

<table>
<thead>
<tr>
<th>Traditional languages</th>
<th>Syntax Based</th>
<th>Structure Based</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOGMA, LOOM,</td>
<td>DAML+OIL</td>
<td>Frame-based</td>
</tr>
<tr>
<td>OntoLingua, KL-One</td>
<td>OIL</td>
<td>DL-based</td>
</tr>
<tr>
<td>F-Logic, PLIB</td>
<td>OWL</td>
<td>FOL-based</td>
</tr>
<tr>
<td>OKBC, RACER</td>
<td>RDF</td>
<td>----</td>
</tr>
<tr>
<td>CycL Language</td>
<td>RDF Schema</td>
<td>----</td>
</tr>
<tr>
<td>KM programming</td>
<td>SHOE</td>
<td>----</td>
</tr>
<tr>
<td>KIF, Common Logic</td>
<td>OIL</td>
<td>----</td>
</tr>
</tbody>
</table>
3.5 ARCHITECTURAL METHODOLOGY

Ontology is a technique which by annotating semantics to provides a basic and common infrastructure for formal resource definitions on the domain of concern. To improve the interest of a learner the Ontology development is useful to provide the learning resource in a well-organized and efficient manner [89]. There was no structured and familiar guiding principle for ontology development till the end of 90’s, and ontology was being seen as an art rather than engineering commotion. Later on, various ontology development methodologies were developed to maintain ontology based management of domain knowledge such as OD-101, Diligent, Methontology, Kactus, and Tove. However, there exist is no established ontology development methodology for knowledge modelling or engineering [90].

Due to the insufficiency of domain ontologies in the area of computer science has to lead to a loss of knowledge in the computer science domain. Hence, there will be a problem faced by the researchers who want efficient information access to meet their objectives or interests. This problem requires an artefact for a solution, and the candidate that could lead to the solution is ontologies that will give a sort of meaning to the terms. The ontologies are having a potential to mitigate the problem by making the concept in the domain of computer science explicit and comprehensible.

One of the objectives of this research is ontology building to provide support to a learner for course personalization is the professional and effectual way of improving the learner’s interest and performance as well. Ontologies are useful for communicating the knowledge related to different educational domains. The ontologies in the domain of CS is inadequate since major ontologies were constructed on other domains such as medical, entertainment, sports, physics and cooking also to provide an appropriate structured knowledge. The ontology related to a specific domain suggests a mutual understanding of the domain concepts and the relationships that exist between them. Mainly the knowledge in computer science domain has not been efficiently exploited by the learners since
the knowledge in CS is dispersed and the possibility to access and reuse has been limited

A well-structured and clearly defined structured methodology helps in reducing the time for ontology development. Since there is no standard methodology for creating course ontology, an architectural elucidation of an iterative process integrated with the Methontology activity states as shown in above Figure 3.4 has been suggested for the development of computer science ontology in this research work.

![Figure 3.4. Ontology Development Process](image)

The intention for developing the ontology for CS domain is to provide a graph based representation of the domain that will be helpful to learners, researchers and other scholars for planning their coursework efficiently. The ontology graphs will provide an insight of the domain and a learner can easily wade through how these courses are related together. The biggest advantage ontology posses are that the focus of the learner will mostly be on the concepts in a domain and the relationships instead of information only. Reusing the ontology can save the time for an expert as they can make use of ontologies that have already been built.
More often the development of ontology is done by an expert group such as ontology engineers, domain expert and pedagogues. Some of the rationale as stated by for Noy et al [91] for building ontologies as follows.

- To study and make the assumptions about the domain Knowledge which are explicit and related with a domain
- The knowledge about the domain should be analyzed properly and shared with software agents and other people
- To enable the reuse of domain knowledge
- The domain Knowledge should be separated from the operational Knowledge

All these points are an exact match with what this study is trying to achieve. The courses in the CS domain are not structured properly rather scattered around. There is no way a learner can find out if courses are overlap between domains because of hard to find out if there is any relation between courses. A short list layed-out by [91] that includes what developing ontology should include is.

- Identifying the classes in the developed ontology
- Classes should be arranged in a taxonomy based hierarchy
- Slots Identification and authorizing values for each slot
- Slot instances should be filled with values

The individuals of a particular class can be created at the time of building the class hierarchy along with the assigned properties of the class. These individuals or instances will support to create an ontology knowledge base but the main problem that needs to be overcome is to build a common standard for creating domain ontologies. This study is concerned with the development of ontology for the courses in the domain of CS such as cloud computing, artificial intelligence, data mining, software engineering and networks. An organized study of certain books in CS, Institute syllabi of CS and also detailed conversation with some expert faculty was done even before classifying the concepts related to a
specific domain. The visualization feature possessed by the ontology and web semantics will have its effect next-gen learning and teaching process

The infantile nature of ontology makes it an attractive research area which can prove a vital technological support to facilitate the processing of information over the web in the form of web indexing, searching, web semantics and even annotations. If ontology is purely developed for the purpose of education it will be a game changer in the direction of creating shareable and reusable educational content for a unified education system. There are some institutes and other groups who have started creating the ontology for educational purpose. This research study also developed CS ontology and create 27 classes and 15 subclasses which are given in Table 3.2 and Table 3.3

Table 3.2 List of the Classes in developed Ontology

<table>
<thead>
<tr>
<th>University Programs</th>
<th>Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>University Programs</td>
<td>Student</td>
</tr>
<tr>
<td>Employee Book</td>
<td>Journal Article</td>
</tr>
<tr>
<td>Domain Artificial Intelligence</td>
<td>Technical Report</td>
</tr>
<tr>
<td>Publication Networking</td>
<td>Conference Paper</td>
</tr>
<tr>
<td>Department Software Engineering</td>
<td>Professor</td>
</tr>
<tr>
<td>Research Cloud Computing</td>
<td>Faculty</td>
</tr>
<tr>
<td>Assistant Professor Data Mining</td>
<td>Associate Professor</td>
</tr>
<tr>
<td>Course Guide</td>
<td>College</td>
</tr>
<tr>
<td>Scholar Administration</td>
<td>Credit</td>
</tr>
</tbody>
</table>

Table 3.3 List of Sub-Classes in developed Ontology

<table>
<thead>
<tr>
<th>Machine Learning</th>
<th>Network Hardware</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intelligent Systems</td>
<td>Fuzzy Mathematics</td>
<td>Big Data</td>
</tr>
<tr>
<td>Knowledge Discovery</td>
<td>Communication</td>
<td>Security</td>
</tr>
<tr>
<td>Software Project Management</td>
<td>Software Testing and Quality Assurance</td>
<td>Formal Methods of Software Engineering</td>
</tr>
<tr>
<td>Natural Language Processing</td>
<td>System Performance Engineering</td>
<td>Theory of Computation</td>
</tr>
</tbody>
</table>
The aim of this study is to realize the Knowledge model is developed for the better of the learner to understand the domain and courses contained in it from a better perspective. The ontologies for various domains exist but the educational context of such ontologies is hidden in terms of quality and the stability. The ontology for various domains and subjects are nonexistent. As a result, the construction of ontology is helpful in dispensing the courses in effectual and in a professional mode and could fruitful in improving learners interest. Noy and McGuiness were well aware of the ontology background and they pen-down few steps that have to be considered during ontology construction. The base of the study is in proposed architectural methodology that was used to configure the development process of the proposed ontology.

3.5.1 Specification:

The goal of this activity state is related with defining reasons that who will be the ontology users, why is being developed and to check out if ontology being developed is already present so that it can be directly reused. The specifications should be comprehensive and concise so that it covers the primary objective of building ontology, its scope and the level of granularity as shown in Table 3.4

Table 3.4. Ontology requirement specification Document for CS Ontology

<table>
<thead>
<tr>
<th>Domain Name</th>
<th>Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>The ontology about the courses in computer science to be used by the learners required during teaching, learning, analysis and so on.</td>
</tr>
<tr>
<td>Level of Formality</td>
<td>Semiformal</td>
</tr>
<tr>
<td>Scope</td>
<td>List of major domains in computer science, sub-domains and the list of courses available</td>
</tr>
<tr>
<td>Source of knowledge</td>
<td>University Syllabus, Books, Faculty Experience</td>
</tr>
</tbody>
</table>
• **Determine scope of the ontology**

To find out the domain and the scope of the ontology being developed, there are some questions to be considered while developing the ontology. The questions are as follows:

What will domain ontology cover?
Who will be the users of ontology?
For what purpose ontology is going to be used?
What questions should ontology answer?

The domain in this study is computer science covering the scope for all the courses meant for graduate, postgraduate and research program, which suggest that all learners’ select among only those courses for the program of their interest. The other related information about the courses is course name, course credits, department who offer the course and the course linked with another course in a same or different program.

• **Consider reuse of existing Ontologies**

Before commencing the ontology creation, a research was conducted to check out whether any organization has developed the ontology in CS domain. After making sure that there is no such ontology available and even if ontology is present and available in a variety of formats, still it can be used with the help of protege. The OntoMerge feature can of protege can interpret any ontology and combine them with the developing ontology.

There are ontologies that exist in different domains but no such ontology in CS domain was present that could have been utilized in this study. The learners’ in the domain of CS has not exploited much knowledge present in it effectively, since most of the knowledge in the CS domain is disbanded and the viability of its access and reuse is very restricted. For this reason and keeping in view the objective of the study the CS ontology was created from the scratch.
3.5.2 Conceptualization

The knowledge scattered in a domain is structured into a conceptual model which helps to understand the problem and the clarification by using domain vocabulary used at the time of ontology specification. Glossary terms are build that includes all domain knowledge regarding the main concepts, properties, verbs and instances. The role is to convert and organize an informal view of the domain into a concept using intermediate representation like the graph notation to provide a conceptual ontology model that developers can understand. These representations help to bridge the gap between the people and how the ontology is formalized as shown in Figure 3.5.

![Figure 3.5 Methontology Tasks for Conceptualization](image)

- **Enumerate essential terms in the ontology**

Before initiating ontology development it is advised that development team should write down vocabulary that is used while ontology is being developed. The practice should be followed prior to the creation of the classes, properties and the relationships and other terms. The motive behind this practice is that during ontology development the terminology list will keep on varying. Thus, observing the queries raised at step 1 of specification task of ontology, the terms that has been extracted and then added in the ontology are course code, course name, and course credits etc. During ontology creation terms that are overlapped were excluded even before moving to the conceptualization phase.
• Define classes and their hierarchy

In protégé, all classes start with the base class Thing from which other classes will spring forth and Thereby super classes of Ontology will be treated as subclases of Thing. In developed ontology Schools, Student, Employee, Research are super classes and will further have subclasses and entities in them. It serves as an umbrella definition for instances in sub-classes. The image of class hierarchy from Protégé is given below in Figure 3.6

![Class Hierarchy in CS Ontology](image)

**Figure 3.6 Class Hierarchy in CS Ontology**

• Superclass University

This class holds all other subclasses in the ontology so that individuals for each subclass can be created. A school has subclasses Department and Programs, Programs, in turn, will contain another subclass which enlists all the programs offered in the ontology, and Research has subclass Publication. The user recommendations will be based on the algorithms and the semantic logic used in the ontology to bind the courses together using some relations. Superclass is displayed in Figure 3.6
• **Superclass Departments**

The Superclass contained a sub class in the form of different programs and departments that offer those programs. The individuals can be created to illustrate a programs being offered in each department. In a potential application that uses this ontology researchers can use DL queries to check what programs that are being offered are and whether they can choose a particular course of interested. The semantic logic developed in the ontology provides information to the learner about the department offering the particular Program. In Figure 3.7 it has been shown the programs that are linked with respective department.

![Figure 3.7 Description Language (DL) Query to Check Programs Offered](image)

- **Superclass Domains**

Super class holds all other major classes and the courses in it in this ontology, shown in below Figure 3.8. The developed ontology consists of domains like Software Engineering, Data Mining, Cloud Computing, Artificial Intelligence, and Networking. The domains are comprised of courses such as language technology, Web services, RFID, intelligent information retrieval, algorithms knowledge discovery technologies, etc. The course can be a part of more than one study program and they are abstracted out of specific program.
3.5.3 Formalization

The major role of this activity is to convert or transform the conceptual model developed in conceptualization activity into a semi-formal or semi-compatible model by using the description logic or frame based system.

- Define the properties of classes and individuals

The individual separation is done using classes by defining the domain and their range. The individual are mutually bounded due to various properties created in the ontology and in the developed ontology the properties are classified in terms of data and object property as shown in Figure 3.9.
Figure 3.10 Query to Check Linked Courses

The properties of an object define the relationships between the individuals of same or different class. Figure 3.10 shows an individual course artificial intelligence and its link with six different courses that might belong to same or different domains with the help of property “is_related” or creating a relationship among the courses. The data property on the other hand contains the information about the individual to which they are assigned without being related to other individuals. All individual courses might have data property CourseName and defining means the real course name and number of credits it posses.

- Define the facets and constraints of the properties:

Table 3.5 and Table 3.6 mentioned below define the properties of the developed domain ontology with their field-name, range, and comments describing their usage. The first table is for object properties, while the second is for data properties. The property is_related, has_course, has_credit, has_deparment are used by the individuals created in their own classes using semantic logic. This suggests that if a new course is introduced in the developed ontology all the above mentioned properties will be automatically added to it. The underlying principle for using is_related property is binding two or more courses together that later will help in course personalization by fetching the related courses with respect to the main recommendations generated.
### Table 3.5 Object properties of the CS Ontology

<table>
<thead>
<tr>
<th>Object property</th>
<th>Field</th>
<th>Range</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has_course</td>
<td>Department</td>
<td>Course</td>
<td>Used on individuals of the department on how many courses they are offering.</td>
</tr>
<tr>
<td>Has_department</td>
<td>Schools</td>
<td>Programs</td>
<td>Used on courses that require which department is offering a particular course.</td>
</tr>
<tr>
<td>Has_credit</td>
<td>Domain</td>
<td>Courses</td>
<td>Used on courses that are having some credits.</td>
</tr>
<tr>
<td>Is related</td>
<td>Domain</td>
<td>Courses</td>
<td>Used on individuals of Courses to which course they are linked with.</td>
</tr>
</tbody>
</table>

### Table 3.6 Defining the data properties of the ontology

<table>
<thead>
<tr>
<th>Data property</th>
<th>Area</th>
<th>Range</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CourseName</td>
<td>Domain</td>
<td>String</td>
<td>The name of a course e.g. Data Mining and Analysis.</td>
</tr>
<tr>
<td>CourseCode</td>
<td>Domain</td>
<td>String</td>
<td>The code of a particular course, e.g. C102</td>
</tr>
<tr>
<td>Credits</td>
<td>Domain</td>
<td>Integer</td>
<td>How many credits a course is worth.</td>
</tr>
</tbody>
</table>
3.5.4 Implementation

This activity requires the use of ontology languages like OWL, RDF schema etc., to build a computable model. The outcome of this activity is a codified ontology in an official language. The ontology development languages should offer an analyzer to guarantee the absence of syntax and lexical errors. It should provide an editor, translator and evaluator to modify the definitions, to allow portability and to detect certain inconsistencies in the ontology. The ontology implementation requires an environment to support the Ontologies selected at the integration phase.

- Creating Individuals / Instances

There are certain steps that need to be followed while creating the an Instance in an ontology and some the steps are as follows (i) A particular class should be selected while creating an individual. (ii) To create an individual in actual sense it should be named properly. (iii) Properties should be added to the individuals. In this research work individually are mostly found in the form of domain courses. The developed ontology has added an approximately 300 courses on the basis of the availability. The main classes domain and department accommodates courses and the department as individuals as per predefined policy.

Figure 3.11 Details of the Individual Course Network Security
In this study the number of objects and the properties used are plenty for developing ontology prototype which essentially includes the semantic logic. The details of individuals of two different classes are present in Figure 3.11 and Figure 3.12 and it can be clearly seen that network security course has 3 credits and is related with mobile and wireless network security course through some relationship. The individual IT department is offering different programs containing various courses.

![Figure 3.12 Details of the Individual Department](image)

3.5.5 Maintenance

This activity helps to correct and update the ontology if required during validation or after implementation phase. Certain activities which are there by default in the maintenance phase of any software product are management activity to activities to support quality control and schedule, while as a support activity helps in documentation, evaluation, integration and knowledge acquisition.

3.6 ONTOLOGY EVALUATION AND VALIDATION

The last phase of ontology development is how the evaluation and validation of the developed ontology can be done as both of these components are important for any development process. The evaluation criterion helps to determine the completeness, and the significance of the ontology being developed while as the process of validation will check whether the developed
ontology is significant and complete and that is the reason Ontology validation, and evaluation is done. The size of the ontology being developed and the community it will serve is used to check the number of efforts needed to validate and evaluate the ontology [92]. Always it is advisable to do ontology validation at the time of its ontology development, even though validation and evaluation can be done after completion of ontology. In this study, the size of the ontology is moderate and is meant for the educational community especially for computer science domain and certain methods were followed to do validation and evaluations the computer science ontology, and the methodology was suggested by [93] which and are given as follows.

3.6.1 Ontology Evaluation and Validation Criteria

The Ontology was evaluated by checking the following criterion:

- Ontology should be Correct (Correctness)
- It should be complete (Completeness)
- It should be effective (Efficacy)
- Minimal redundancy (Uniqueness)
- Easy to maintain (Maintainability)

The Ontology validation is done by checking the following criterion:

- The ontology has to be pertinent to the field of study (Relevancy)
- The purpose of ontology development need to be fulfilled (Purpose)
- Significant examples should be created
- The examples shown should be explicable (Suitable)
- There should be consistency in the ontology
- The ontology need to be clear, robust, concise, and complete (quality)

3.7 CONCLUSION

The chapter starts with perception of ontology and then narrow down its concept in the domain of CS. The chapter provides a realistic view of the ontology along with its usability and significance other than the idea alone. The reuse of
ontology, its sharing, continuity and the kind of support it will provide for personalization has been considered in this study. Even though most of the concepts in CS have already been scanned, yet there are certain other concepts that can be added further. It would possibly be more interesting to give expansion to the knowledge model by linking the concepts from CS domain with other domains and then use it for some particular applications. The intent to provide a knowledge model for the domain of computer science is for the better understanding of the domain by the learners. The model also helps to put forward the idea that ontology can fill the gap between amorphous courses in a domain and the explicit knowledge of contemporary classification. The study made non-obvious attempt to associate the sub-domains from various domains by matching them using semantics. The study has few achievements some of them are as follows

- The architectural methodology for developing the ontology was proposed and in addition to that computer science domain ontology was developed.

- The strategy for representing the development of domain model and the support it will provide for course personalization.

- The ontology resulted from CS domain covering five domains such as data mining, software engineering, networking, Cloud computing, and artificial intelligence and about fifteen sub-domains.

- The courses are formalized in hierarchy along with the relationship that interconnect the courses from all domains and sub-domains and describing a adequate computer science domain model.

- The iterative nature of the methodology used for developing ontology was crucial for obtaining the better results.

The protege tool was used to model the knowledge about CS domain by generating the automatic OWL code while creating the ontology. Post development, the ontology was checked for consistency by validating and
evaluating it with the help of an inbuilt reasoner known as FACT++. Thus, an effort is being made in present research work to frame domain ontology for CS for supporting e-learning personalization feature using RS approach. The ontology developed consists of major domain and sub-domains and will be helpful for learners to study their domain in a much consistent way. The ontology developed is fed to the RS to support in generating more recommendations by checking the relationship among the courses which are actually generated using the learner input query and in this way ontology can enhance the performance and satisfaction level of the learner.

The knowledge representation of a domain based on ontology will be much robust that it makes the developing framework visible by screening the abstract composition of a domain. Thus, in current study the ontology development for the computer science domain will be fruitful for modern system of education where it will serve as knowledge model and assist in personalizing an e-learning system. The ontology created will be helpful for amalgamating the institutes by providing them a common policy for the effective utilization of sharing the resource. Most of the domain courses in CS are covered in the present thesis with some of the limitations as a future work. The noticeable one is to spread the knowledge model among other educational domains.