ABSTRACT
Software requirement engineering is an important process in software development. When considered software development as a whole, 75% of software failures are due to impaired software requirements. The proposed technique involves a multi viewed approach comprising of Controlled Natural language and ontologies that can be used for representing the requirements. Ontology is an explicit information modelling method which can be used to model applications and their interactions. Controlled natural languages are subsets of natural languages, obtained by restricting the grammar and vocabulary in order to reduce or eliminate ambiguity and complexity that enable reliable automatic semantic analysis of a language. The ontologies are constructed based upon the semantic similarities between the domain and requirement models. The CNL is developed by restricting the vocabulary based on certain rules. The similarity between the two representations can be defined by a program that extracts the objects and relationship between them. Tool based verification of the similarities is performed. This approach is applicable in software development areas and many official applications such as banking system, currency conversion, weather reports, transport, and sports.

Keywords
Requirements Engineering , Ontology , Controlled Natural Language

1. INTRODUCTION
Software engineering (SE) is the engineering discipline through which software is developed. Commonly the process involves finding out what the client wants, composing this in a list of requirements, designing an architecture capable of supporting all of the requirements, designing, coding, testing and integrating the separate parts, testing the whole,
deploying and maintaining the software. Programming is only a small part of software engineering.

Requirements engineering (RE) being the first phase of software engineering deals with the process of discovering, eliciting, documenting and maintaining the requirements for a particular computer-based system. Computer systems are designed, and anything that is designed has an intended purpose. If a computer system is unsatisfactory, it is because the system was designed without an adequate understanding of its purpose, or the purpose is deviated from the intended one. Both problems can be mitigated by careful analysis of purpose throughout a system’s life. Requirements Engineering provides a framework for understanding the purpose of a system and the contexts in which it will be used. The requirements engineering bridges the gap between an initial vague recognition that there is some problem to which there is a computer technology, and the task of building a system to address the problem.

Ontology formally represents knowledge as a set of concepts within a domain, and the relationships between pairs of concepts[3]. It can be used to model a domain and support reasoning about entities. An ontology renders shared vocabulary and taxonomy which models a domain with the definition of objects/concepts, as well as their properties and relations. Ontologies are the structural frameworks for organizing information and are used in artificial intelligence, the Semantic Web, systems engineering, software engineering, informatics, library, enterprise bookmarking, and information architecture as a form of knowledge representation about the world or some part of it [5]. The creation of domain ontologies is also fundamental to the definition and use of an enterprise architecture framework. Model for describing the world that consists of a set of types, properties, and relationship types. There is also generally a view that the feature of the model in an ontology closely resembles the real world.

There are many problems associated with requirements engineering. The nature of problems includes defining the system scope, problems of understanding and problems of volatility. The major problems are sharing equal understanding among the different communities involved in the development of a given system, and problems in dealing with the volatile nature of requirements. Problems of understanding during elicitation can lead to requirements which are ambiguous, incomplete, inconsistent, and even incorrect. If changes are not accommodated, the original requirements set will become incomplete, inconsistent with the new situation, and potentially unusable because they capture information that has since become obsolete. The requirement traceability issue is also a major factor concerned with the requirement engineering. These problems may lead to poor
requirements and the cancellation of system development, or else the development of a system that is later judged unsatisfactory or unacceptable, has high maintenance costs, or undergoes frequent changes. By improving requirements elicitation, the requirements engineering process can be improved, resulting in enhanced system requirements and potentially a much better system. In order to overcome these issues, a model integrating controlled natural language and ontology providing a multi-viewed approach is proposed to represent requirements from different perspectives of all the stakeholders and the development team.

2. RELATED WORK

2.1 A scenario-driven approach to trace dependency analysis
Egyed et al. proposed an approach to trace dependency analysis. Software development artifacts such as model descriptions, diagrammatic languages, abstract (formal) specifications, and source code are highly interrelated where changes in some of them affect others [2]. Trace dependencies characterize such relationships abstractly. The research area focused on an automated approach to generating and validating trace dependencies. The research addressed the severe problem that the absence of trace information or the uncertainty of its correctness limits the usefulness of software models during software development. This is considered to be an important issue affecting requirements engineering. This approach also proposed a method that automates what is normally a time consuming and costly activity due to the quadratic explosion of potential trace dependencies between development artifacts.

2.2 Issues in requirement elicitation
Christel M and Kang k., in their research paper titled “Issues in requirements elicitation”, clearly listed all the issues that affect the requirement elicitation phase of software development. According to their research, the requirement elicitation phase suffers from issues dealing with the improper detailing of scope of the system. The communication between the various people involved in developing the system also is causing major problems, since requirement elicitation highly involves with collection of information related to the system. The next major issue listed deals with the volatile nature of the requirements. The paper implies that if these problems are not taken seriously, it might lead to a poor requirement and also the end result may be a failure.

2.3 Communication problems in requirements engineering
Al-Rawas et al. explains about the problems of communication between disparate communities involved in the requirements specification activities [1]. The requirements engineering phase of software development projects is characterized by the intensity and importance of communication activities. During this phase, the various stakeholders must be able to communicate
their requirements to the analysts, and the analysts need to be able to communicate the specifications they generate back to the stakeholders for validation. The results of this study are discussed in terms of their relation to three major communication barriers: ineffectiveness of the current communication channels, restrictions on expressiveness imposed by notations, social and organizational barriers. The results confirm that organizational and social issues have great influence on the effectiveness of communication. They also show that in general, end-users find the notations used by software practitioners to model their requirements difficult to understand and validate.

2.4 Revisiting ontology-based requirements engineering in the age of the semantic web systems
Dobson G, Sawyer P, in “Revisiting ontology-based requirements engineering in the age of the semantic web”, propose usage of dependability ontology compliant with the IFIP Working Group 10.4 taxonomy and discuss how this, and other ontologies, must interact in the course of Dependability Requirements Engineering. In particular usage of the links between the dependability ontology, ontology for requirements and domain ontologies, identifying the advantages and difficulties involved are discussed in detail. Ontology is generally based upon some logical formalism, and has the benefits for requirements of explicitly modeling domain knowledge in a machine interpretable way, e.g. allowing requirements to be traced and checked for consistency by an inference engine, and software specifications to be derived. With the emergence of the semantic web, the interest in ontologies for Requirements Engineering is on the increase. Lots of research has been concentrated upon re-interpreting software engineering techniques for the semantic web. Usage of ontology is proved to be highly beneficial for requirement engineering processes.

3. METHODOLOGY
In software development requirement refinement is an important process. It has a tremendous impact on all its software development phases. Though a lot of research is involved in the area of solving the various issues affecting the requirement engineering, few issues remain unsolved. The common issues that haunt the requirement elicitation phase include problems concerned with areas of scope, volatility, communication, traceability [4]. Defining the scope of the system plays the major role in developing any new system. Hence solving the scope issues is highly necessary. The requirements of the customers are prone to changes throughout the development of the system which is generally referred as the volatility issue. Developing a system involves a wide class of people and hence
communication plays a major role. Any fault that arises in developing the system as a result of mistaken communication is referred as communication issues. Any issues that affect describing and following a requirement in both forward and backward direction are referred as traceability issues. In order to overcome these issues a system consisting of a multi-viewed approach is proposed. The system integrates two views for representing the perspective of different stakeholders. Ontologies and controlled natural languages are used for this purpose. Thus, the proposed technique provides a way for dealing with the various issues affecting requirement engineering.

In this paper, we propose a comprehensive approach for dealing with the various issues affecting the requirement elicitation process. The requirement document is analyzed and split into individual tokens. The tokens are used to extract only the required relevant terms involved in the document. The tokens are used as a base for construction of the ontology. The requirement document is next represented in terms of controlled natural language. The controlled natural language representation also extracts only the key terms involved. Both the representations show the relationship among the objects that are extracted. The similarities between the two representations are measured to verify the level of accuracy.

![Overall process of integrating ontology and CNL](image)

**Figure 1. Overall process of integrating ontology and CNL**

Figure 1 represents the process involved in designing the multi-viewed approach consisting of ontology and control natural language.

The ontological view of requirements document involves representing the requirement document in terms of concepts and relations among them. The process also involves tagging of the requirement in prior to the construction of the ontology.
The CNL view of requirements involves representing the document through restricted vocabulary. This representation enables to clearly establish the relationships among the identified objects through a restricted vocabulary. The mapping of the ontology representation and the CNL representation is performed with the help of v-doc and GMO algorithm. The level of similarity is noted.

3.1 Ontological view of requirement

The requirement document is represented in terms of ontology. Ontology formally represents knowledge as a set of concepts within a domain, and the relationships between pairs of concepts. Ontologies are used to model a domain and support reasoning about entities. Ontology provides shared vocabulary and taxonomy which models a domain with the definition of objects/concepts, as well as their properties and relations.

Ontologies are the structural frameworks for organizing information and are used in artificial intelligence, the Semantic Web, systems engineering, software engineering, informatics, library, enterprise bookmarking, and information architecture as a form of knowledge representation about the world or some part of it. The creation of domain ontologies is also fundamental to the definition and use of an enterprise architecture framework. Model for describing the world that consists of a set of types, properties, and relationship types. There is also generally a view that the feature of the model in ontology closely resembles the real world.

The first step in representing the requirement document as ontology may require tagging the parts of speech in the document. The tagging is performed with the help of a tool called POS Tagger. The POS Tagger does the process of marking up a word in a text (corpus) as corresponding to a particular part of speech. Once the tagging is completed, the ontology can be constructed. Formally it can be said that ontology is a statement of a logical theory. Ontologies are often equated with taxonomic hierarchies of classes without class definition and the subsumption relation.

Ontologies need not to be limited to these forms. Ontologies are also not limited to conservative definitions, that is, definitions in the traditional logic sense that only introduce terminology and do not add any knowledge about the world. To specify a conceptualization, one needs to state axioms that do constrain the possible interpretations for the defined terms. The construction of ontology starts with identifying the concepts. Concept represents a set of entities within a domain. Relations are constructed next and it specifies the interaction among concepts involved.

Ontology of a program is obtained by defining a set of representational terms. In such ontology, definitions associate the names of entities in the universe of discourse (e.g. classes, relations, functions, or other objects).
with human readable text describing what the names mean, and formal axioms that constraint the interpretation and well-formed use of these terms. The ontologies can be designed on the basis of domain document. This consists of 3 steps:

1. Tokenization is done by Stanford pos tagger which reads the text document and assign the parts of speech to each word such as noun, verb, adjectives, etc. (figure 2)

2. The isolation of individual tokens shows the number of nouns, verbs, adjectives in the domain document.

3. Based on the Nouns, verbs and adjectives, the concepts involved are identified.

4. Ontology can be created by using the protégé tool based on the concepts identified. (figure 3)
3.2 CNL view of requirements
A sample requirement document is considered. The requirement document will be in ordinary English which has no rules and semantics to make it understandable by the system we convert that into controlled natural language. CNL refers to controlled natural language that are subsets of natural languages, obtained by restricting the grammar and vocabulary in order to reduce or eliminate ambiguity and complexity that enable reliable automatic semantic analysis of a language A tool called Fluent Editor is used to provide a platform for constructing the required CNL.

We use Controlled English as the knowledge modeling language. Supported by a suitable Editor, it prohibits one from entering any sentence that is grammatically or morphologically incorrect and actively helps in correcting any error. The Controlled English is a subset of Standard English with restricted grammar and vocabulary; in order to reduce the ambiguity and complexity inherent in full English. The relationships between the various objects involved in the domain are established.

A taxonomy tree is also displayed representing the objects hierarchically. A taxonomy tree is constructed with four important parts. They are “thing” which shows “is a” relationship between concepts and relations. The “nothing” shows concepts that cannot have any instances. The “relations” that shows hierarchy of information about the relation between concepts and/or the instances. And finally “attributes” that shows the hierarchy of attributes. Modal expressions are constructed using must, should, can, must not, should not, and cannot. Similarly complex expressions can also be constructed. There are 4 steps:
1. The domain document is represented in terms of Controlled English which is a subset of Standard English with restricted grammar and vocabulary in order to reduce the ambiguity and complexity inherent in full English. (Fig 4)
2. The CNL phrases for every corresponding OWL statements are created using a CNL editor.
3. Modality is enabled to create modal expressions that are used to express relationship in CNL.
4. Complex sentences as well as simple sentences can be created using the CNL editor.

Figure 4. Construction of CNL

3.3 Mapping of ontology and CNL
The mapping of the CNL and ontology are performed, after both representations are constructed completely. The objective of this module is to measure the level of similarity that is established between the two forms of representation. The obtained CNL is extracted as objects and relationships. The obtained objects are written on to a separate file. The objects obtained from ontology are kept on a separate file manually. The mapping of the two representations are performed through a program as well as verified by a tool. The program is used to extract the objects and the relationship that exist between them. The first part of program deals the extraction of the objects and relationship between the objects. The program
consists of predefined relationships. The extracted objects and relationships from the OWL file is compared with the predefined relationships that are already entered. After extraction, the mapping of the extracted data is performed. If the objects and relationship match, a positive output is obtained stating “ontology matching”. If the objects and relationships don’t match, a similar output stating “ontology not matching” is obtained. The mapping involves the usage of V-Doc algorithm and the GMO algorithm. The v-doc constructs virtual documents for each entity in the ontologies and use Vector Space Model to compute similarities between the virtual documents. The GMO algorithm is a novel graph matching algorithm. The mapping of the ontology representation and the CNL representation is performed with the help of v-doc and GMO algorithm. The level of similarity is noted. The steps are

1. Relationship and objects that exists in the ontological view of representation is extracted.
2. Relationship and objects that exists in the CNL view of representation is extracted.
3. The extracted relationships and objects are placed in individual files.
4. A program is used to check the level of similarity between the objects that are extracted.
5. Tool based verification is further performed.

![Figure 5. Matching OWL files](image-url)
Thus, the tool based verification of matching is shown if Figure 5. The ontology1 column shows the list of objects in OWL file for ontology. The ontology2 column shows the list of objects in OWL file for CNL. The similarity column shows the level of similarity between the two OWL files.

4. CONCLUSION
As for as software development is concerned, requirement elicitation is an important process. The existing system used single view approach, thus failing to deal with the lot of issues affecting requirements engineering. Therefore a technique is proposed to refine the requirements by integrating ontology and controlled natural language, thus providing a multi viewed approach. The ontological view is mapped with the controlled natural language view to calculate the level of similarity. Thus, usage of a multi viewed approach helps in resolving most of the issues affecting the requirement elicitation process. The proposed process provides a simple and useful Traceability scheme. Refining the requirements provides a lot of advantages and reduces a lot of cost in building the system. The research on the area of RE has grown fast in the last few years. In spite of this fact, there are still open issues. In our work we initially identified such issues and investigated the main existent initiatives that are addressing them. Further, the work can be improvised by exploring the full potential of ontologies, thus improving quality of the knowledge base.

5. REFERENCES