CHAPTER 6

Conclusion

- Summary
- Findings and observations
- Limitations and future work
- References
6.1 Summary
The present work attempts to demonstrate the generation of semantic information services using the emerging Semantic Web technologies in the application area of eLearning. Typically many users of the web search engines experience with noisy and context-less information retrieval. We often end up with more irrelevant documents in retrieved sets instead of relevant document sets. We often spend a significant amount of time in retrieving the required information on the Web even form retrieved sets. There is no semantic agent available to work on our behalf, so that without spending much time we can retrieve our all required and relevant information. It is understandable that machines do not understand the information on the Web. The information on the Web is mainly encoded in human understandable and presentable form and consumable by the humans only. Interpretation by the machines is difficult and often not possible. The obvious reason for this is lack of structuring and describing the information on the Web in a machine understandable form.

In order to provide semantic based information services, it is important to represent the information on the Web in machine understandable form. In this process, the objective should be to describe the domain knowledge in machine processable way and also to describe the services in a machine processable form using the ontology languages. The aim is to develop a model for cross-domain applications. In this regard, the following hypotheses are put forward.

1. Domain knowledge base can be developed using ontology languages
2. The web services can be developed using ontology languages and
3. Semantic web services have cross-domain applications.

In order to keep the work focused and to demonstrate semantic based information system, a set of possible application domains are thought of, such as, eLearning, eGovernance, eHealth, eCommerce, etc. Finally, eLearning is chosen as a possible application domain to set the scope of the
study and implementation of the system. Further, in order to keep the work focused a methodology as discussed in chapter 1 is adopted.

Since eLearning is the chosen application domain, the first task is to understand the concept of eLearning and then to come out with a working definition of it. The working definition of eLearning, “an interactive learning mode in which the learning content is available online and provides feedback to the student’s learning activities that can be structured and presented in user defined personal manner” (discussed in chapter 2). Another important concept, which is equally important, is the “learning object”. The working definition of learning object is enumerated as “it is a digital and web based self-contained learning resource having specific learning objective and re-usable to support learning” (discussed in chapter 2). The characteristics of an ideal eLearning system are described in chapter 2. The issues with the present eLearning systems are discussed in chapter 5.

The architecture adapted places learner requirement in a central position in the present system. Hence the identification of the learners’ characteristics is essential, to make the system learner-friendly and to provide with semantic access to the system are discussed in chapter 1, which further discusses the learning management systems and the open source software, the evolution of the eLearning. The most important aspect of this section is to show how technologies played an important role in the evolution process of eLearning instead of just looking at them from the perspective of ‘which technology’ is used.

The next step is to evaluate the applicable existing metadata standards. This study has a big impact on our present work as the vision is achieving semantic interoperability. In the present Web in most of the cases, software agents are incapable of communicating with each other or understanding each other. And as a result they cannot compile the information available in different sources and finally provide the consolidated information to the users. This is because of lack in consensus among the indexers. In most of
the cases indexers prefer to develop and use their own descriptors in describing the resources instead of adapting available standards. It is always preferable to use the existing metadata standards (either world standards or de facto ones) if they serve our purpose. Only in cases where it is absolutely required and inevitable, the standard sets could be extended further. There are many other reasons why we need metadata standards as listed in chapter 3.

Dublin Core metadata standard in conjunction with IEEE LOM metadata standard, which was specially developed for describing the learning objects is used in the present work. Similarly, vCard, a metadata standard in encoding the personal information is used to describe the learners in conjunction with the IMS Learner Information Package, which was specially designed for describing the learners. An enhanced version of mapping between the Dublin Core and IEEE LOM is presented in chapter 3.

The next step was to make an extensive study of the present Semantic Web technologies and tools (discussed in chapter 4), as one of the goals of present work is to apply the emerging Semantic Web technologies, tools and techniques to build a system. It is worth mentioning that while studying semantic technologies and tools, importance is given to the open source tools as it is also the goal to use only the open source tools in developing the present system. A list of (discussed in chapter 4) the present issues with web services are such as,

1. At present resources and services are not in machine understandable form, these are in human understandable form.
2. The representation of resources and services on the Web are unstructured and they are only loosely related to each other.
3. Searching resources and services on the Web at present is keyword based; no semantics of the resources are used.
4. Interoperability between toolkits is a problem due lack of consensus in vocabularies used and standards for description.
Chapter 6: Conclusion

The new emerging Semantic Web and the Semantic Web technologies are discussed. Semantic Web, propounded by Tim Berners-Lee is described as, a vision of the next generation web, which enables web applications to automatically collect web documents from diverse sources, integrate and process information and interoperate with other applications in order to execute sophisticated tasks for humans [1], where the key motivations are, automatic information processing on the Web by providing the machine-friendly information, achieving the interoperability, so that the software agents can work in collaboration and to make the computers work on behalf of us more instead of using them just as a tool. The Semantic Web technologies are discussed in the line of layered approach to Semantic Web as originally discussed by Tim Berners-Lee in [1]. The idea of layered approach is to ensure that we achieve the consensus at least in small steps at a time. Different technologies are applied in each of the layers as discussed in chapter 4. URI (Uniform Resource Identifier) is at the bottom of the layers. One important aspect of Semantic Web is linked data. In this extension of the present Web, it is envisioned that every bit of data will be linked each other using the URI. Every piece of data on the Web will have its unique identifier. So, it will be easy for machines to trace and share resources wherever they are stored. The next layer talks about XML and XML Schema. XML language is used to structure the information in a tree, subsumption form. It makes possible for machine to process the data, but of course it does not talk about the semantics, which is important for machines to process and retrieve the information semantically. It is particularly useful in exchanging documents on the Web. Whereas, XML Schema acts as a grammar or we can say it is used for validation of XML document.

XML Schema deals with the data types and minimum level of data constraints. In order to improve the data further on the Web, the next layer is about RDF and RDFS. RDF is not a language rather it is a data model. It models the data in a triplet format, in a similar format as we have in natural language, following the order, subject, predicate and object. RDFS is another important aspect, considered though it has limited capacity in
expressing and storing the semantics. The layer on top of it is OWL Ontology layer, which is much more powerful, more expressive, and logic driven. Since it is logic driven, it allows us to infer knowledge from the existing facts. The intent of OWL language is to provide additional machine-processable semantics for resources, that is, to make the machine representations of resources more closely resemble their intended real world counterparts [2]. And finally at the top of the layers is the trust layer, which is to prove the inferred knowledge. Chapter 4 describes the description logic and the available languages based upon the description logic.

Semantic Web Services (SWS), a combination of web services with Semantic Web technology is proposed to overcome the lack of web services in terms of semantic description that would facilitate the semantic discovery and integration. In this regard, various technologies of SWS, emphasizing OWL-S and WSMO, the two key technologies, their advantages, the differences and the issues involved within them are discussed (chapter 4). The N3Logic, a subset of First Order Logic (FOL) like Description Logic (DL) is presented as it permits rules to be integrated with RDF. It provides certain built-in functions, which allow information to be used from the Web and reasoned with. The goal of N3Logic is to make a minimal extension to the RDF data model, which makes it possible to use the same language for rules and data, enabling the representation of the rules and data using the same language. N3Logic provides simplicity and completeness [3]. Chapter 4 discussed the relationship of N3Logic with RDF, RDFS and OWL.

Chapter 4 also discusses the reasoners and the reasoning tools. A reasoner (semantic) is a piece of software able to infer logical consequences from a set of asserted facts or axioms. The semantic reasoner provides a richer set of mechanisms in order to infer new facts from a set of asserted facts. There are many reasoning engines available; few also as OSS. For example, CWM, Pellet, Racer, Fact, Fact++, Euler, Racer, Jess, XSB and so on. The present work uses Euler. These reasoning engines are grouped mainly into
two based upon the reasoning methodologies used in executing the rules. They are,

1. *Forward chaining*: starts with the available data and uses inference rules to extract more data (for example, user provided data) until a goal is reached. Following forward chaining mechanism, the inference engine searches the inference rules until it finds one where the antecedent (*If* clause) is known to be true. And once it is found, the reasoner can conclude or infer, the consequent (*Then* clause), and in result infer a new information (facts); and

2. *Backward chaining*: it is an inference method used in automated theorem proofs, proof assistants and other artificial intelligence applications. Backward chaining starts with a list of goals (or a hypothesis) and works backwards from the consequent to the antecedent to see if there is data available that will support any of these consequents.

The system architecture and the implemented system are presented in chapter 5, which address some of the following existing issues in the present eLearning system, such as:

1. *Lack of group and personalized learning spaces*
2. *Presentation of the entire learning material instead of relevant information actually sought by a learner*
3. *Fixed or pre-ordered inflexible Learning sequence.*
4. *No Reusability of resources*
5. *Lack of semantic interoperability*
6. *Quality assurance and reliability of the retrieved learning materials.*
7. *Lack of Relevance Ranking*

We enlisted the system perspectives (chapter 5) in terms of the student, and the tutors and/or administrators point of view. We discussed on ontology
Chapter 6: Conclusion

and eLearning perspectives. We presented a conceptual framework of semantic driven eLearning system, which we named as “semantic learning layer cake”. It is developed in a bottom-up layered approach. It is developed considering the three important issues, such as, content, context and structure (chapter 5). These are in general applicable in context of online information retrieval system.

Chapter 5 presents the system architecture. It also presents the ontology frameworks developed in order to build our system. Ontologies are the main backbone of the present system. There are mainly three ontologies, such as,

1. **Domain ontology**- Domain ontology models the concepts and their relationships of a particular subject area to be taught instead the general concepts. Wikipedia [4] defines domain ontology as, it is a domain-specific ontology, models a specific domain, or part of the world. It represents the particular meanings of terms as they apply to that domain, in the present work, it is ‘indexing and abstracting’. Particularly in an eLearning scenario domain ontology describes the course contents semantically as it is built in poly-hierarchical structure with more formalized relations between the subject concepts.

2. **Document ontology**- Document ontology is to model and to describe the learning contents and their component parts semantically. It is envisioned to simulate the learning object internal structure (discussed in details in chapter 5, section 5.8.2.1) as we see in real world into an ontological form for semantic computation. It comprises a set of classes and sub-classes, semantic relations (object properties) and attributes (data-type properties).

3. **Student ontology**- Student ontology is designed to describe the students’ community requirement. It captures the students’ information. In general we can call it as ontology based learners’
profile. It is an important component for any ontology driven eLearning system, as one of our mottos is to develop the learner centric (personalized) eLearning system. It is being considered as a background knowledge provider about a student to the system. Student ontology is extremely important in service point of view while we look for satisfying the learners’ learning goals more meaningfully (contextually).

The framework is provided as UML diagrams for the above mentioned three main ontologies. Protégé is used as an ontology editor for building and validating the ontologies (chapter 5). These ontologies are formalized in OWL-DL.

After developing the ontologies the system is populated with data. It is worth mentioning that in demonstrating the system, 'Indexing and Abstracting' is selected as the learning course under the programme of Library and Information Science. Though the system uses the course from Library and Information Science, the model is expected to accommodate any other courses from the other domains, for example, Mathematics, Physics, Geography, History, etc. because the system has adopted the ontology framework, general structure and content of formal domains.

As Indexing and Abstracting as a learning course, domain ontology for the same in a faceted structure is developed. SKOS is used in developing domain ontology primarily and is extended further to suit our particular requirements (chapter 5).

The system is populated both with document ontology creating the lessons for the Indexing and Abstracting course and also the student ontology by creating some data for students.

The services and logic rules are designed for manipulating the data and services. Addition of logic rules is to make inferences, choose courses of
action and answer questions. It is worth to point here that, one has to be careful in choosing the logic, as it should be powerful enough in describing complex properties of objects, but again not too much expressive, so that the execution cost become high in terms of execution time and inefficiency in reasoning the knowledge. Considering all these things, the N3Logic, a subset of the First Order Language (FOL) is adopted, as N3Logic is expressive enough and is useful as a tool in the open Web environment [5].

The present work demonstrates seven different types of cases as mentioned below (details discussed in chapter 5). These cases were designed based upon some of the important aspects discussed in system perspectives section 5.3 of chapter 5. The following use cases are dealt with

Case 1: Learning style (student type)
Case 2: Cognitive learning style
Case 3: Background knowledge
Case 4: Education level
Case 5: Learners language and learning resource language
Case 6: Students available network connection speed and the learning materials required network connection speed
Case 7: Network connection speed: provision of alternative version

In the following sections, present the observations. Though some of the objectives of building and demonstrating semantic eLearning services are achieved, there are certain limitations as presented below.

6.2 Findings and observations
1. Semantic Web technologies have the potential to improve the present state of the web based information retrieval system. Key products enroute are ontologies and ontology languages. These languages are rich in both ways, syntactically and semantically. These ontology languages are logically enriched and enable machines to infer the knowledge semantically.
2. It is also observed that to resolve issues like, personalization, reuse of the resources, etc. we need to provide more fine-grained description of the content.

3. Achieving semantic interoperability is also possible. It is possible to achieve by adopting the metadata standards that are already popular without having to reinvent the wheel. The present system can share the knowledge with other systems configured with the same metadata standards.

4. The existing semantic tools and techniques are efficient and sufficient to start with at least at local level in developing the system in order to provide enriched and contextual information services.

5. The area of semantic operations in digital repositories is akin to the demonstrated system discussed here.

6.3 Limitations and future work

6.3.1 Limitations

1. The present work dealt with limited types of data. But variety of data can be expanded by domains and types of resources, which may pose unseen challenges.

2. The present work deals with a course only in higher education i.e. Undergraduate, Graduate courses. This can be extended by considering the others, for example, primary level, secondary level, basic level, etc. where the user profiles may be significantly different.

3. Multilinguality issues are to be addressed even though this is not deemed in scope of the present work.

4. Scalability is not tested, as the present system is more a proof of concept.

6.3.2 Future work

1. Full-fledged modular ontology framework for eLearning domains—the present work demonstrates the role of formal ontologies in removing the constraints of the current eLearning system which also
explores the efficiency and effectiveness of modular based framework. But this work is carried out within a limited scope. The present work can be further investigated by developing a full-fledged modular ontology framework for different eLearning domains.

2. Visualization is an important aspect, which demands its exploration - it is an important perspective, which can be further explored.
3. Quality assurance, is an important issue in eLearning domain, needs to be explored. This is not only applicable for eLearning but in general it is needed to be investigated through surveys and studies.
4. Extending this work for the physically challenged users - the present study was conducted considering only the physically normal users, but there is lot of potential in extending this work for the physically challenged users as highlighted in section 5.9 (case 7) of chapter 5.

6.4 References
Appendix A – Application profile for Domain Ontology

The following are the description of classes, sub-classes, properties and constraints used in domain ontology.

1. Classes and Sub-classes Used in the Domain Ontology
Following classes and sub-classes are used in building the domain ontology.

1.1 skos:Concept: identifies the domain concepts. The notion of skos:Concept class is to describe the conceptual or intellectual structure of a knowledge organization system or to refer the specific ideas or meanings within a KOS [1]. For example, it identifies the concepts, such as, Indexing, Coordinate indexing, Pre-coordinate indexing, Post-coordinate indexing, etc. of a domain, called, Indexing and abstracting.

1.2 skos:ConceptScheme: identifies the concept scheme (e.g., Colon Classification, DDC, UDC, LCSH, MeSH) used in identifying the domain concepts. But it is also possible to create and use concepts as stand-alone entities. In our case, we used standard KOS, such as, CC, DDC, UDC and LCSH in identifying the domain concepts. Since the concepts under domain Indexing and abstracting are not available extensively in any of the above mentioned KOS, we decided to derive them. In this regard we used text-books, glossaries, etc. on indexing and abstracting in finding and extracting the domain concepts.

2. Semantic relations and Attributes used in Domain Ontology
Following are the semantic relations and attributes used in building the domain ontology.

2.1 skos:hasTopConcept: associates the concept instances of skos:Concept and points to which skos:ConceptScheme the concept belongs to. The domain and range of this property are skos:Concept and skos:ConceptScheme respectively.
2.2 skos:topConceptOf: is used as inverse property of skos:hasTopConcept. The domain and range of this property are skos:ConceptScheme and skos:Concept respectively.

2.3 skos:broader: is a semantic relation that connects a concept to its broader concept in skos:Concept class. The meaning of the relation between the linked concepts is inherent in the link property. The domain and range of this property are the same as skos:Concept.

2.4 skos:narrower: is a semantic relation that connects a concept to its narrower concept in skos:Concept class. The meaning of the relation between the linked concepts is inherent in the link property. skos:narrower is the inverse property of the semantic relation skos:broader. The domain and range of this property are the same, skos:Concept.

Both the properties, namely, skos:broader and skos:narrower mainly represent the hierarchical relationships between the concepts. We can also say that they act alike “is-a” relation, and these relations preserve the semantics.

2.5 skos:related: is a semantic relation connect two associative concepts of skos:Concept class. The meaning of the relation between the linked concepts is inherent in the link property. The domain and range of this property are the same as skos:Concept.

2.6 skos:hiddenLabel: it assigns the "hidden" labels for a concept. The “hidden” labels are useful when a user is interacting with a knowledge organization system via a text-based search function. The user may, for example, enter misspelled words when trying to find a relevant concept. The misspelled query can be matched against a "hidden" label, in that case the user will be able to find the relevant concept, but the "hidden" label will not otherwise be visible to the user (so further mistakes are not encouraged) [1]. The domain and range of this property are skos:Concept and xsd:string respectively.
2.7 skos:altLabel (alternate label): is useful in assigning an alternate label of a concept. The domain and range of this property are skos:Concept and xsd:string respectively.

2.8 skos:prefLabel (preferred label): is useful in assigning a preferred lexical label of a concept. The domain and range of this property are skos:Concept and xsd:string respectively.

It is worth to point here that the above three labels are extensions of rdfs:label. These labels provide clues to the meaning of a concept.

2.9 skos:note : provides information about the concepts, for example, definition, example, scope note, etc. These notes can be used in several ways, such as to generate the documentation, instructions and so on. SKOS extended the property skos:note into its sub-properties, such as, skos:definition, skos:example, skos:scopeNote, skos:historyNote, skos:changeNote and skos:editorialNote. The domain and range of these properties are skos:Concept and xsd:string respectively.

2.10 skos:definition : provides a complete explanation of the intended meaning of a concept.

2.11 skos:example : provides an example of the use of a concept.

2.12 skos:scopeNote : provides information about the intended meaning of a concept, especially as an indication of how the use of a concept is limited in indexing practice.

2.13 skos:historyNote: portrays significant changes to the meaning or the form of a concept.

2.14 skos:changeNote: documents fine-grained changes to a concept, for the purposes of administration and maintenance.
2.15 *skos:editorialNote*: provides information that is an aid to administrative housekeeping, for example, reminders of editorial work still to be done, or warnings in the event that future editorial changes might be made.

In addition to the SKOS properties, the following properties are defined locally to design a rich ontological model for domain ontology.

2.16 *skos-ext:hasInstance*: provides the instance of a domain concept. For example, POPSI, PRECIS, etc. are kind-of (instances) pre-coordinate indexing system. The domain and range of this property are the same as *skos:Concept*. It is a transitive property.

2.17 *skos-ext:isInstanceOf*: It is a inverse property of *skos-ext:hasInstance*. The domain and range of this property are the same as *skos:Concept*.

2.18 *skos-ext:synonym*: represents the synonymous term of a concept. The domain and range of this property are skos:Concept and xsd:string respectively.

3. References

Appendix B – Application profile for Document Ontology

1. Classes and Sub-classes Used in Document Ontology

The following classes and sub-classes are used in modeling the document ontology.

1.1 Entity: this class category identifies information about the entities, such as, learning object, course, program and the people, organizations contributing to the learning objects. It consists of four sub-classes, namely, Program, Course, Learning Object, and Contribute as discussed below.

1.1.1 Program: it represents a knowledge area about which the system is providing customized courses. It is an approved sequence of units in related areas. For example, applicants seeking admission to the program Masters of Science in Library and Information Science (MS-LIS) are required to select a ‘major sequence’ in a discipline offered by the faculty.

The following are the constraints applied on class Program. It implies that maximum four keywords can be created in describing the program.

```
lom-gen:keyword max 4
```

1.1.2 Course: a course is a sequence of the topics tailored to a program. It identifies information about the courses offered by an educational establishment.

The following is the constraints applied on class Course. It implies that maximum four keywords can be created to describe the courses.

```
lom-gen:keyword max 4
```

1.1.3 LearningObject: a learning object is a resource; digital or non-digital (we mainly dealt with digital objects in our present work), which can be used, re-used or referenced during technology supported learning, for example, computer-based training systems, interactive learning
environments, intelligent computer-aided instruction systems, distance learning systems, collaborative learning environments [1].

Examples of Learning Objects include multimedia content, instructional content, learning objectives, instructional software and software tools, and persons, organizations, or events referenced during technology supported learning.

This class category identifies information about the learning objects. The following constraints are applied on class Learning Object and on its sub-classes as discussed below.

\[
\text{dc:subject min 1}
\]

It implies that minimum one subject proposition must be chosen in describing the learning materials (which comes from the instances of class skos:Concept).

\[
\text{lom-gen:keyword max 4}
\]

It implies that teachers are allowed in creating maximum four keywords in addition to the minimum one standard subject proposition which comes from the instances of class skos:Concept as mentioned above in describing the learning materials.

This class category consists of three sub-classes, such as, Topic, ContentUnit and SupportResource as discussed below.

1.1.3.1 Topic : a learning object representing a theoretical explanation about a topic of knowledge that is complete in the scope of the intended explanation. It can be subdivided in subtopics explaining specific parts of the topic explanation on a deeper level of knowledge or specificity. A topic can be suitable for all courses customizing a discipline or only for some of
them. This class category identifies the bibliographic information, such as, title, subject, language, education level, intended end user role, etc. of the topics. The item specific information is identified by the class category Content Unit as discussed below.

The following constraints implies that the value of predicate educationLevel for topics belong to the value set of class lom-edu:Context.

\[
\text{dcterms:educationLevel} \text{ only } \text{lom-edu:Context}
\]

Similarly, the following one implies that the predicate value of predicate lom-edu:learningResourceType is exactly one.

\[
\text{lom-edu:learningResourceType exactly 1}
\]

Learning objects covering the topics can be in two different forms, such as, Atom and Module.

1.1.3.1.1 Atom : it is to identify information of the simplest learning materials which deals with a single "learning concept". For example, a HTML page deals with a single learning concept "Java:Inheritance". It may also have some item(s), such as, figure, table, image, etc. as parts defining the same learning concept. This is mainly applicable where it deals with a very short communication of a concept.

1.1.3.1.2 Module : it is to identify information of complex learning objects consisting of several chapters or sections or sub-sections or both. For example, Book, Book chapters, Journal articles, a single Web page (or HTML file) that might incorporate one or more sections, images. etc. It might also include .ZIP and other packages that can be accessed as individual files, where those files together form a single aggregate resource. It is further classified into the following three sub-classes, such as,
1.1.3.2 ContentUnit: this class category is to identify the item related information. By item related information we mean the information very much particular to each learning object. For example, id number, location, format, etc. of a learning object. The class category is further divided into the following five sub-classes as shown below.

1.1.3.2.1 Dataset: it identifies item information of learning objects when the content is dataset, i.e., a collection of data. Dataset can be in the form of database, lists and tables. This class category consists of three sub-classes, such as,

1.1.3.2.1.1 Database: this class is to identify item information of the content when content is in the form of database.

1.1.3.2.1.2 Lists: is to identify information about the ordered content (list) of a learning object.

1.1.3.2.1.3 Tables: is to identify information when content is in a tabular form.

1.1.3.2.2 Image: It is to identify item information of the learning objects when the content is an image. For example, images and photographs of physical objects, paintings, prints, drawings, other images and graphics, animations and moving pictures, film, diagrams, maps, musical notation.

This class category consists of two classes, MovingImage and StillImage.

1.1.3.2.2.1 MovingImages: this class category is to identify item information when the contents are moving images. It consists the following sub-classes,
1.1.3.2.2.1 Animations
1.1.3.2.2.1.2 TelevisionPrograms
1.1.3.2.2.1.3 Videos

1.1.3.2.2 StillImages: this class category is further divided into five sub-classes, such as,

1.1.3.2.2.2.1 Drawings
1.1.3.2.2.2.2 GraphicDesigns
1.1.3.2.2.2.3 Maps
1.1.3.2.2.2.4 Paintings
1.1.3.2.2.2.5 Plans

1.1.3.2.3 Software: class identifies item information of the learning object when the learning content is any software tool; examples include a Java, C source file, MS-Windows .exe executable, Perl script, etc.

1.1.3.2.4 Sound: is to identify item information when the learning content is in a audio form. For example, music playback files format, an audio compact disc, and recorded speech or sounds. This class category is divided further into two, Music and Speech.

1.1.3.2.4.1 Music
1.1.3.2.4.2 Speech

1.1.3.2.5 Text: this class is to identify item information when the learning content is in a textual form. For example, books, letters, dissertations, poems, newspapers, articles, archives of mailing lists.

1.1.3.3 SupportResource: this class category is to identify information about the learning resources recommended by tutors. It is further divided into three according to their genre, such as, Document, Links and Tool.
1.1.3.3.1 Document: this class category is to identify information about the documents, such as, books, articles (journal, newspaper articles), slide presentations, etc. It is further divided into the following classes as shown below,

- 1.1.3.3.1.1 Book
- 1.1.3.3.1.2 SlidePresentation
- 1.1.3.3.1.3 Article
  - 1.1.3.3.1.3.1 JournalArticle
  - 1.1.3.3.1.3.2 NewspaperArticle

1.1.3.3.2 Links: identifies the link information of the learning resources suggested by the tutors.

1.1.3.3.3 Tool: this class category is to identify information when the suggested materials are some software tools. It is further classified into a class, called, Software.

1.1.4 lom-life:Contribute: those entities (i.e., people, organizations) that have contributed to the state of this learning object during its life cycle (e.g., creation, edits, publication). This category consists of two sub-classes, such as,

- 1.1.4.1 lom-life:Entity: the identification of and information about entities (i.e. people, organizations) contributing to this learning object. The entities shall be ordered as most relevant first.

1.1.4.2 lom-life:Role: kind of contribution. It is worth to note here that, minimum the author(s) of the learning object should be mentioned. The possible values that could be assigned for this class are, such as, author, publisher, validator, editor, graphical designer, technical implementer,
content provider, technical validator, educational validator, script writer, instructional designer, unknown, etc. [2]

1.2 LearningObjectives: this specifies the learning objectives of the learning materials. It is worth to note the value of this class yields the learning context. In our model we defined the following as learning objectives

<table>
<thead>
<tr>
<th>analysis</th>
<th>develop</th>
</tr>
</thead>
<tbody>
<tr>
<td>apply</td>
<td>estimate</td>
</tr>
<tr>
<td>cause</td>
<td>evaluate</td>
</tr>
<tr>
<td>classify</td>
<td>evidence</td>
</tr>
<tr>
<td>compare</td>
<td>introduce</td>
</tr>
<tr>
<td>define</td>
<td>measure</td>
</tr>
<tr>
<td>demonstrate</td>
<td>procedure</td>
</tr>
<tr>
<td>describe</td>
<td>solution</td>
</tr>
<tr>
<td>design</td>
<td>synthesize</td>
</tr>
<tr>
<td>determine</td>
<td></td>
</tr>
</tbody>
</table>

1.3 dc:Language: this is to identify information of the resource language. The values are enumerated following ISO 639-2 [3].

1.4 lom-edu:Context: a class of entity specifying 'for who' the resource is intended or useful. It is the principle environment within which the learning and use of this learning object is intended to take place [1]. This class consists a sub-class, Higher education. But it can be further extended, refined according to the purpose of the system.

Suggested values are, Graduate and Undergraduate. But any other values can also be considered depending upon the requirements.

1.5 lom-edu:Difficulty: is to identify information how hard it is to work with or assimilate the learning object for the typical intended target
audience. Suggested values are, \textit{easy}, \textit{very easy}, \textit{medium}, \textit{difficult} and \textit{very difficult}.

1.6 \textit{lom-edu:IntendedEndUserRole} : identifies information about the principle user(s) for which this learning object was designed [1]. For example, a \textit{learner} works with a learning object in order to learn something; an \textit{author} creates or publishes a learning object; a manager manages the delivery of the learning objects, say, typically a curriculum, etc.

1.7 \textit{lom-edu:LearningResourceType} : this category is to identify information of type of the learning resources. It indicates mainly potential educational use(s) of content associated with the learning resource.

It consists of two sub-classes, Education resource and Examination resource.


1.7.2 \textit{ExaminationResource}: value space includes, \textit{exam}, \textit{project task}, \textit{questionnaire} and \textit{self-assessment}.

1.8 \textit{lom-tech:Format} : technical datatype(s) of (all the components of) learning object. This data element is used to identify the software needed to access the learning object. It is also used to indicate the resource medium to end-users. [1]

1.9 \textit{Scheme} : this class category identifies information about the scheme, such as language and concept scheme used. It consist of two sub-classes, such as, Language scheme and \textit{skos:ConceptScheme} (discussed in Appendix A)
Appendix B – Application profile for Document Ontology

1.91 LanguageScheme: the language scheme used to define the language of the learning resources and the students language. For example, language scheme, like, ISO 639-1, ISO 639-2, etc. are popular. In our model we used ISO 639-2 [3].

2. Properties Used in Document Ontology
The properties used in document ontology along with the property restrictions are presented in tabular form as shown below.

<table>
<thead>
<tr>
<th>Defined property</th>
<th>Domain</th>
<th>Range</th>
<th>Inverse property</th>
<th>Logical characteristics</th>
<th>Const raints</th>
</tr>
</thead>
<tbody>
<tr>
<td>lom-edu:learningResourceType</td>
<td>Topic</td>
<td>lom-edu:LearningResourceType</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dc:language</td>
<td>Entity</td>
<td>dc:Language</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dcterms:educationLevel</td>
<td>Topic Support Resource</td>
<td>lom-edu:Context</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>isContentUnitOf</td>
<td>Content Unit Topic ContentUnit</td>
<td>hasContentUnit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>isPartOf</td>
<td>Topic</td>
<td>Topic</td>
<td>hasPart Transitive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>isRecommendedBy</td>
<td>Support Resource</td>
<td>lom-life:Entity</td>
<td>Recommends</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dc:contributor</td>
<td>Learning Object</td>
<td>lom-life:Entity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>isVersion</td>
<td>Content</td>
<td>Content</td>
<td>hasVers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nOf</td>
<td>Unit</td>
<td>tUnit</td>
<td>ion</td>
<td>Transitive</td>
<td></td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------</td>
<td>-------</td>
<td>---------------------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>isAvailableToCourse</td>
<td>Topic</td>
<td>Course</td>
<td>courseHasPart</td>
<td>Transitive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Program</td>
<td>Course</td>
<td>Tailors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>isTailoredBy</td>
<td>lom-life:Entity</td>
<td>LearningObject</td>
<td>dc:creator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>isCreatorOf</td>
<td>dc:Language</td>
<td>LanguageScheme</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>courseHasPrerequisite</td>
<td>Course</td>
<td>Transitive</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LearningObject</td>
<td>LearningObject</td>
<td>hasPrerequisite</td>
<td>Transitive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LearningObjectives</td>
<td>ContentUnit</td>
<td>hasLearningObjective</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lom-edu:difficulty</td>
<td>LearningObject</td>
<td>lom-edu:Difficulty</td>
<td></td>
<td>Cardinality = 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Course</td>
<td>lom-edu:Difficulty</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LearningObject</td>
<td>lom-edu:IntendedUserRole</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LearningObject</td>
<td>lom-edu:UserRole</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ContentUnit</td>
<td>lom-tech:Format</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Learning</td>
<td>Skos:C</td>
<td></td>
<td>Min</td>
<td></td>
</tr>
</tbody>
</table>
### Appendix B – Application profile for Document Ontology

<table>
<thead>
<tr>
<th>ct</th>
<th>Object</th>
<th>Concept</th>
<th>Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>lom-edu:typicalLearning Time</td>
<td>Learning Object</td>
<td>xsd:string</td>
<td>1</td>
</tr>
<tr>
<td>lom-tech:location</td>
<td>Content Unit</td>
<td>xsd:string</td>
<td></td>
</tr>
<tr>
<td>Duratio n</td>
<td>Learning Object</td>
<td>xsd:string</td>
<td></td>
</tr>
<tr>
<td>lom-edu:typicalAgeRange</td>
<td>Learning Object</td>
<td>xsd:string</td>
<td></td>
</tr>
<tr>
<td>dc:title</td>
<td>Learning Object</td>
<td>xsd:string</td>
<td></td>
</tr>
<tr>
<td>require NCSpeed</td>
<td>Content Unit</td>
<td>xsd:string</td>
<td>Functional</td>
</tr>
<tr>
<td>lom-tech:size</td>
<td>Learning Object</td>
<td>xsd:string</td>
<td></td>
</tr>
<tr>
<td>dc:description</td>
<td>Entity</td>
<td>xsd:string</td>
<td></td>
</tr>
<tr>
<td>lom-gen:keyword</td>
<td>Learning Object</td>
<td>xsd:string</td>
<td>Max. cardinality = 4</td>
</tr>
<tr>
<td>hasPrerequisite</td>
<td>Learning Object</td>
<td>xsd:string</td>
<td></td>
</tr>
</tbody>
</table>
3. References


Appendix C – Application profile for Student Ontology

1. Classes and Sub-classes Used in the Student Ontology
The following class and sub-classes are used in modeling the domain ontology.

1.1 stu-onto:StudentType : defines student learning style (details discussed in Chapter 2). We define four types of students such as, Reading-writing learners, Audio learners, Visual learners and Kinesthetic/tactile learners. These are modeled in the ontology as the instances for this class.

1.2 stu-onto:CognitiveLearningStyle : specifies the cognitive learning style of a learner. The cognitive characteristics formalize the type of information processing and reasoning the student uses. We identified five types of cognitive learning style, such as, Analogue-Analytic, Concrete-Generic, Deductive-Evaluative, Indefinite_style, and Relational-Synthetic [1]. These are modeled in our ontology as instances for this class.

1.3 stu-onto:Gender : this is to define the students gender information. We defined two values for this class, namely, Male and Female.

1.4 stu-onto:EducationLevel : this class category identifies information relating to the learners educational level. It has a sub-class, called, HigherEducation as discussed below.

1.4.1 stu-onto:HigherEducation : specifies the information about the higher education level. We defined two values for this class, such as, Undergraduate and Graduate.

1.5 stu-onto:Student : this is the central class of the student ontology. It identifies the students and all the other related information about each individual student.

Following are some of the constraints assigned on class Student.
It implies that, the value of predicate `educationLevel` for a student (say, student_1, an instance of class `Student`) belongs to the value set of class `EducationLevel`.

It implies that, the value of predicate `educationLevel` of a student is exactly one. It will not allow creating more than one value for the predicate `educationLevel`.

By assigning this, system restricts in defining value for the predicate `hasCognitiveLearningStyle` only from the value set of class `CognitiveLearningStyle`.

It restricts us in assigning the value for predicate `gender` of a student into one only.

Similarly, it also restricts us in assigning value for predicate `typeOfStudent` from the value set of class `StudentType` for a student.

1.6 `ims-lip:Activity` : this class category identifies any learning-related activity in any state of completion and it could be self-reported. It includes formal and informal education, training, work experience, and military or civil service [2].
1.7 *ims-lip:Affiliation* : this class category identifies the students' professional associations and the associated roles. This class has a sub-class, called, *Organization* as discussed below.

1.7.1 *stu-onto:Organization* : identifies the information about the organization with which a student is associated.

1.8 *ims-lip:Competency* : identifies the competencies of a student.

1.9 *ims-lip:Interest* : this class identifies the hobbies and recreational interests of the learner. For example, painting, story writing, etc.

1.10 *ims-lip:QCL* : this class category identifies information of the students educational qualification, certifications and licenses. It has a sub-class, called, *Qualification* as discussed below.

1.10.1 *stu-onto:Qualification* : specifies the qualification related information of the students.

In addition to the above classes, our ontology consists of another class, named as, *Time*. It is used more as a common class applicable to all the modular ontologies. It is mainly used to serve the purposes like to identify the students year of passing, the course duration, etc. This class consists of 4 sub-classes, such as, *Day*, *Month*, *Year* and Season. It is worth to mention that we did not exploit this class much and could be carried out in our future work.

2. **Properties Used in Student Ontology**

The properties used in document ontology along with the property restrictions are presented in tabular form as shown below.
<table>
<thead>
<tr>
<th>Property</th>
<th>Domain</th>
<th>Range</th>
<th>Inverse property</th>
<th>Logical characteristics</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>stu-onto:gender</td>
<td>stu-onto:Student</td>
<td>stu-onto:Gender</td>
<td></td>
<td></td>
<td>Functional</td>
</tr>
<tr>
<td>stu-onto:hasInterest</td>
<td>stu-onto:Student</td>
<td>ims-lip:Interest</td>
<td>stu-onto:isInterestOf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stu-onto:cognitiveLearningStyleOf</td>
<td>stu-onto:CognitiveLearningStyle</td>
<td>stu-onto:Student</td>
<td>stu-onto:hasCognitiveLearningStyle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stu-onto:degreeObtainedFrom</td>
<td>stu-onto:Student</td>
<td>stu-onto:Organization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stu-onto:registeredForCourse</td>
<td>stu-onto:Student</td>
<td>Course</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stu-onto:languageEfficiency</td>
<td>Stu-onto:Student</td>
<td>dc:Language</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stu-onto:typeOfStudent</td>
<td>stu-onto:Student</td>
<td>stu-onto:StudentType</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stu-onto:languageProficiency</td>
<td>stu-onto:Student</td>
<td>dc:Language</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stu-onto:languageProficiency</td>
<td>stu-onto:Student</td>
<td>dc:Language</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Resource</td>
<td>Data Type</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------------------</td>
<td>-------------</td>
<td>--------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>onto:degreeGrantedBy</td>
<td>onto:Qualification</td>
<td>onto:Organisation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stu-onto:languageDisability</td>
<td>stu-onto:Student</td>
<td>dc:Language</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ims-lip:competency</td>
<td>stu-onto:Student</td>
<td>ims-lip:Competency</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stu-onto:hasQualification</td>
<td>stu-onto:Student</td>
<td>stu-onto:Qualification</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stu-onto:affiliation</td>
<td>stu-onto:Student</td>
<td>stu-onto:Organisation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stu-onto:educationLevel</td>
<td>stu-onto:Student</td>
<td>stu-onto:EducationLevel</td>
<td></td>
<td>Cardinality = 1</td>
<td></td>
</tr>
<tr>
<td>stu-onto:yearOfPassing</td>
<td>stu-onto:Student</td>
<td>Time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stu-onto:hasKnowledgeOn</td>
<td>stu-onto:Student</td>
<td>xsd:string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stu-onto:password</td>
<td>stu-onto:Student</td>
<td>xsd:string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vcard:orgUnit</td>
<td>stu-onto:Student</td>
<td>xsd:string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vcard:orgName</td>
<td>stu-onto:Student</td>
<td>xsd:string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------</td>
<td>------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vcard:country</td>
<td>stu-onto:Student</td>
<td>xsd:string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vcard:locality</td>
<td>stu-onto:Student</td>
<td>xsd:string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vcard:pcode</td>
<td>stu-onto:Student</td>
<td>xsd:string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vcard:street</td>
<td>stu-onto:Student</td>
<td>xsd:string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stu-onto:student_id</td>
<td>stu-onto:Student</td>
<td>xsd:string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stu-onto:hasNCSpeed</td>
<td>stu-onto:Student</td>
<td>xsd:string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vcard:EMAIL</td>
<td>stu-onto:Student</td>
<td>xsd:string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vcard:BDATE</td>
<td>stu-onto:Student</td>
<td>xsd:string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vcard:FN</td>
<td>stu-onto:Student</td>
<td>xsd:string</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix C – Application profile for Student Ontology

<table>
<thead>
<tr>
<th>dc:description</th>
<th>onto:Qualification</th>
<th>ims-lip:Interest</th>
<th>ims-lip:Competency</th>
<th>xsd:string</th>
</tr>
</thead>
</table>

3. References

Bibliography


42. CiteSeerX. http://citeseerx.ist.psu.edu/


45. CWM. http://www.w3.org/2000/10/swap/doc/cwm


63. Dublin Core Education Community. http://dublincore.org/groups/education/
64. Dublin Core Metadata Element Set, version 1.1. http://dublincore.org/documents/dces/
74. ELearning. http://en.wikipedia.org/wiki/ELearning_2.0#ELearning_2.0
75. Elena. http://www.elena-project.org/
77. Emerald: http://www.emeraldinsight.com
91. GNU operating system. http://www.gnu.org
118. IMS Global Learning Consortium, IMS Learning Resource Meta-
119. IMS Learner Information Package Best Practice & Implementation
120. IMS Learning Resource Meta-Data Best Practice and
http://www.imsproject.org/metadata/imsmdv1p2p1/imsmd_bestv1p2p1.htm
121. IMS Learning Resource Meta-Data Best Practice Guide for IEEE
Specification.
122. Introduction to SKOS. http://www.w3.org/2004/02/skos/intro
Assembly of Personalized Learning Content on the Semantic Web. Lecture
Notes in Computer Science, 4011, pp. 545-559.
automatic annotation of learning content. International Journal on Semantic
Web and Information Systems, 2(2), pp. 91-119.
129. Jovanovic, J., Gasevic, D., Brooks, C., Eap, T., Devedzic, V.,
providing educational feedback, Spector J. M., Sampson D. G., Okamoto
T., Kinshuk, Cerri S. A., Ueno M. and Kashihara A. (Eds.), In Proceedings
of the 7th IEEE International Conference on Advanced learning


156. MathML. http://www.w3.org/Math/


161. MIME Media Types. http://www.iana.org/assignments/media-types/


175. Ontology.
http://en.wikipedia.org/wiki/Ontology_(information_science)
179. OWL-S 1.0 Release. http://www.daml.org/services/owl-s/1.0/
180. OWL-S 1.1 Release. http://www.daml.org/services/owl-s/1.1/
209. SHOE, simple HTML ontology extensions FAQ. http://www.cs.umd.edu/projects/plus/SHOE/faq.html#q1.1


236. W3C Semantic Web activity. www.w3.org/2001/sw/
248. XSB. http://xsb.sourceforge.net/index.html